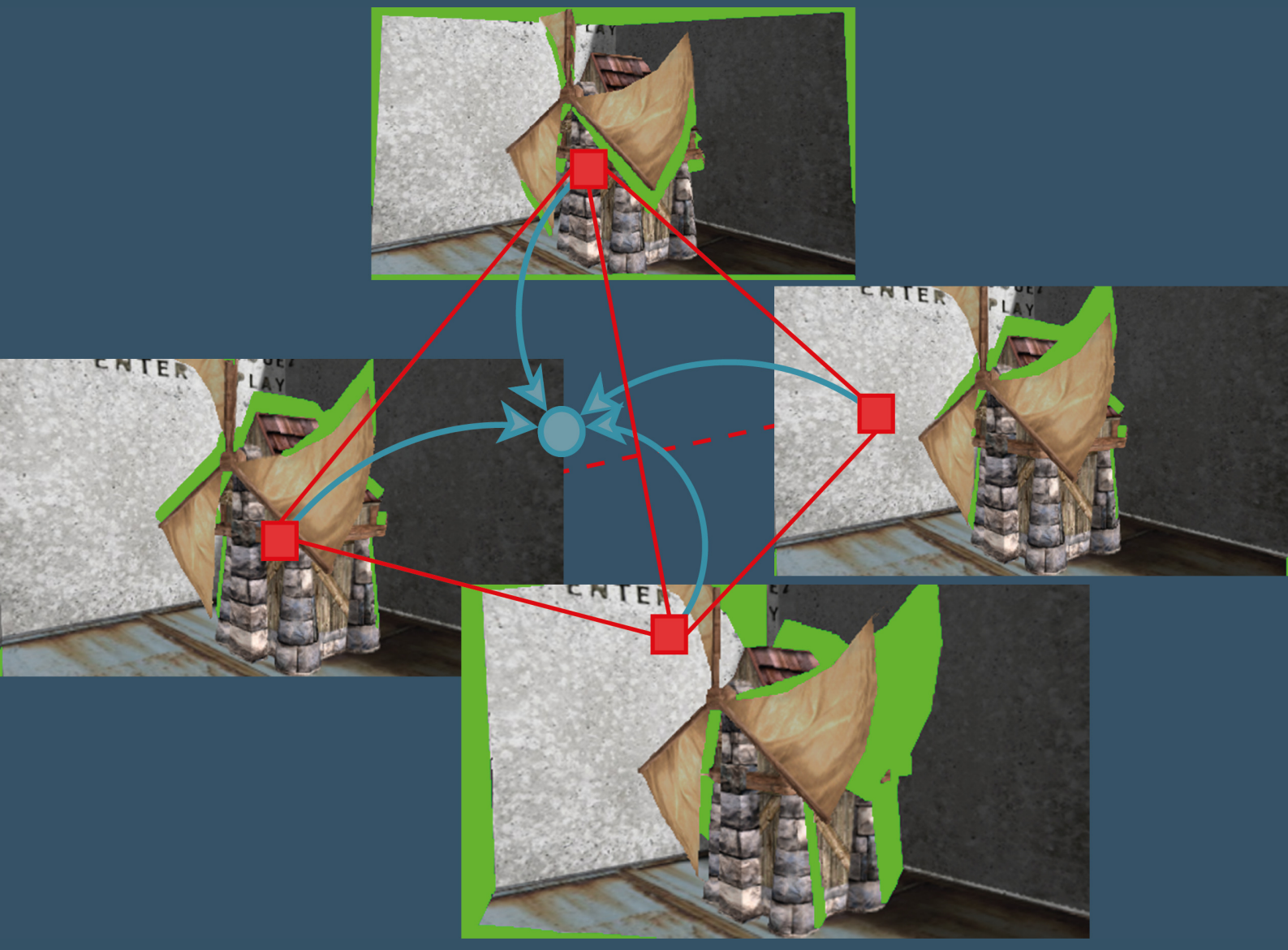
Christian Linz

Correspondence Estimation and Image Interpolation for Photo-Realistic Rendering



Correspondence Estimation and Image Interpolation for Photo-Realistic Rendering

Von der Carl-Friedrich-Gauß Fakultät

Technische Universität Carola-Wilhelmina zu Braunschweig

zur Erlangung des Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte

Dissertation

von Christian Linz

geboren in Zweibrücken

am 1. Juli 1979

Eingereicht am: 02.12.2010

Disputation am: 29.04.2011

Referent: Prof. Dr.-Ing. Marcus Magnor

Koreferent: Prof. Dr.-Ing. Philipp Slusallek

(2011)

Abstract

Free-viewpoint video is a new form of visual medium that has received considerable attention in the last 10 years. Most systems reconstruct the geometry of the scene, thus restricting themselves to synchronized multi-view footage and Lambertian scenes. In this thesis we follow a different approach and describe contributions to a purely image-based end-to-end system operating on sparse, unsynchronized multi-view footage. In particular, we focus on dense correspondence estimation and synthesis of in-between views. In contrast to previous approaches, our correspondence estimation is specifically tailored to the needs of image interpolation; our multi-image interpolation technique advances the state-of-the-art by disposing the conventional blending step. Both algorithms are put to work in an image-based free-viewpoint video system and we demonstrate their applicability to space-time visual effects production as well as to stereoscopic content creation.

Kurzfassung

3D-Video mit Blickpunktnavigation ist eine neues digitales Medium welchem die Forschung in den letzten 10 Jahren viel Aufmerksamkeit gewidmet hat. Die meisten Verfahren rekonstruieren dabei die Szenengeometrie und schränken sich somit auf Lambertsche Szenen und synchron aufgenommene Eingabedaten ein. In dieser Dissertation beschreiben wir Beiträge zu einem rein bild-basierten System welches auf unsynchronisierten Eingabevideos arbeitet. Unser Fokus liegt dabei auf der Schätzung dichter Korrespondenzkarten und auf der Synthese von Zwischenbildern. Im Gegensatz zu bisherigen Verfahren ist unser Ansatz der Korrespondenzschätzung auf die Bedürfnisse der Bilderinterpolation ausgerichtet; unsere Zwischenbildsynthese verzichtet auf das Überblenden der Eingabebilder zu Gunsten der Lösung eines Labelingproblems. Das resultierende System eignet sich sowohl zur Produktion räumlich-zeitlicher Spezialeffekte als auch zur Erzeugung stereoskopischer Videosequenzen.

Zusammenfassung

Visuelle Effekte sind heutzutage aus Kinofilmen nicht mehr wegzudenken. Die Stereoskopie hat in den letzten zwei Jahren ebenfalls eine Renaissance erlebt und wird in naher Zukunft mit 3D Displays auch Einzug in die Wohnzimmer halten. Dadurch entsteht eine erhöhte Nachfrage nach Inhalten für diese neue Form von Anzeigen. Die Produktion von visuellen Effekten und stereoskopischen Inhalten setzt dabei gegenwärtig auf teure und aufwendige Aufnahmetechniken mit oftmals hunderten von Kameras respektive auf eine komplette Modellierung in 3D und auf computergenerierte Animationen.

Am Institut für Computergraphik wurde mit der "virtuellen Videokamera" ein System entwickelt, welches einen wesentlich kostengünstigeren Ansatz verfolgt und eine interaktive Erkundung einer Szene auf Basis einiger weniger unsynchronisierter Videoströme erlaubt. Die Ausgangsbasis für das System bildet ein Algorithmus zur Zwischenbildsynthese welcher auf Erkenntnissen der Wahrnehmungspsychologie basiert. Der Algorithmus versucht dabei wichtige Kantenstrukturen in den Bildern in Übereinstimmung zu bringen und gleichförmige Flächen konsistent zu bewegen. Das interpolierte Bild entsteht durch räumlich adaptives Blenden der verformten Eingangsbilder. Dieses Verfahren birgt zwei Probleme. Zum einen funktioniert der kantenbasierte Ansatz zur Korrespondenzschätzung nur auf Szenen mit moderater Kantenkomplexität, Szenen mit komplexen Hintergründen lassen das Verfahren scheitern. Zum anderen stellt das adaptive Blenden eine Tiefpassfilterung des Bildes dar, was dazu führt dass die interpolierten Bilder weich gezeichnet wirken und Details verloren gehen. In dieser Dissertation gehen wir beide Probleme an. Im ersten Teil evaluieren wir den Stand der Technik in der Korrespondenzschätzung im Hinblick auf die Eignung für die virtuelle Videokamera und leiten aus den Erkenntnissen ein Verfahren zur Korrespondenzschätzung ab, welches speziell auf die Anforderungen der Zwischenbildsynthese ausgerichtet ist. Im Speziellen erlaubt unser Verfahren die robuste Schätzung von symmetrischen Korrespondenzen über große

Pixeldistanzen ohne die Anwendung einer Auflösungspyramide. Im zweiten Teil ersetzen wir das adaptive Blenden der verformten Bilder durch die Lösung eines Labelingproblems. In Kombination mit unseren symmetrischen Korrespondenzfeldern sind wir in der Lage hohe Frequenzen zu erhalten und damit schärfere interpolierte Bilder zu erzeugen. Im dritten Teil bringen wir die entwickelten Verfahren in der virtuellen Videokamera zur Anwendung. Wir zeigen dass sich die virtuelle Videokamera sowohl zur Produktion räumlich-zeitlicher Spezialeffekte als auch zur Erzeugung stereoskopischer Sequenzen eignet, ohne dabei auf teure Hardware bzw. aufwendige Aufbauten zurückzugreifen. Die virtuelle Videokamera schlägt damit eine Brücke zwischen Laborexperiment und realer Filmproduktion.

Acknowledgements

I am very thankful for the excellent working conditions I was provided with at the Computer Graphics Lab of the Technical University of Braunschweig. In particular, my gratitude goes to my supervisor Marcus Magnor, who guided and supported my research. Thanks also to my friends and fellow researchers for helpful scientific discussions and first and foremost lots of fun, both during serious research and in pursuit of less serious matters. I would especially like to thank Timo Stich, Christian Lipski, Anita Sellent, Kai Berger, Lorenz Rogge and Felix Klose for working on several publications with me. Thank you, Anja, for making the administrative part of my work as easy as possible! Special thanks to Anita, Kai and Christian for proof-reading parts of this thesis.

Finally, I'm most grateful to my family for all their patience and support during the last years, always encouraging me to put forth my studies. Anne, thank you for your encouragement and motivation throughout the last months - thanks for being there for me!

1

Introduction

1.1 Motivation

Nowadays, immersive entertainment plays a major role and is starting to outperform traditional media. Experiencing a revival in 2008, stereoscopic cinema has started its triumphant success and its market share has increased from 2% in 2008 to 11% in 2009. Following this trend, stereoscopic television sets are finding their way into living rooms, market researchers predict a market share of 37% in 2014. Another popular example for immersive entertainment is the analysis of sports events with changing viewpoints and virtual augmentations, as recently used in broadcasts of the soccer world cup. However, in both examples the observer takes on a passive role, being able to watch the immersive content but not to interact with it. Full interaction, such as a flexible choice of viewpoint would be extremely beneficial for sports broadcasts and educational documentaries.

A number of researchers are researching on free-viewpoint video (FVV) and 3D television with the goal to obtain a streamable medium that gives the user more control over aspects of playback, in particular choice of viewpoint. Most of the research in this direction has concentrated on reconstructing a full three-dimensional description of the scene which then serves as basis for the rendition of arbitrary viewpoints. Despite in-depth research, this approach to 3D television still faces several hurdles. First of all, a full 3D reconstruction is only feasible for Lambertian scenes, and often a complete scene reconstruction is not possible due to visibility constraints from a limited number of cameras. Further on, almost all algorithms for 3D reconstruction require synchro-

nized and precisely calibrated acquisition hardware, allowing capture only in controlled environments.

This thesis investigates a purely image-based approach to 3D video and aims at providing some of the necessary tools required for such a system. We first introduce the individual components of such a system, i.e. data acquisition, preprocessing, navigation space construction and rendering. We then discuss two major aspects of an image-based free-viewpoint navigation system: high-quality dense correspondence field estimation, and ghosting-free synthesis of in-between views. Putting it all together, we show how such a system and the developed algorithms can be used for intuitive, inexpensive visual effects design in a post-production stage and for the creation of real-world stereoscopic footage.

1.2 Main Contributions

Figure 1.1: Visual effects and stereoscopic output synthesized with the approaches discussed in this thesis.

Throughout the course of this dissertation, parts have already been presented at various conferences and published in conference proceedings and journals [88, 89, 90, 92]. These publications are the foundation of this thesis which incorporates them under the framework of a purely image-based space-time navigation system and presents further studies and improvements. The main contributions are:

- A novel approach to estimate dense correspondence fields using multi-exposure images, Ch. 4. Instead of using high-speed cameras to capture fast motion events, we propose to use multiply exposed images which project several instants of a motion sequence onto a common image plane. Together with a single exposure

shortly before or after the motion sequence, we are able to derive dense motion fields for every single time instant captured in the multi-exposure image [92].

- The evaluation of the current state-of-the-art in dense correspondence estimation for the use in an image-based free-viewpoint navigation system, Ch. 5. We evaluate four state-of-the-art algorithms with respect to their numerical quality as well as with respect to the perceptual quality of the interpolation sequence created from the computed flow fields [91]. This evaluation study serves as the basis for the derivation of a high-quality long-range correspondence estimation, tailored to the needs of image interpolation, in Ch. 6.

- The proposal of a symmetric, long-range optical flow formulation based on dense SIFT descriptors, Ch. 6. Compared to state-of-the-art algorithms, the proposed approach is more robust for image pairs featuring large pixel displacements and also yields better interpolation results [88, 89].

- A novel formulation of multi-image interpolation as a labeling problem, Ch. 7. Instead of adaptively blending several images to obtain the interpolated image, we propose to solve a labeling problem that determines a single source image for every pixel in the interpolated image. With our approach, we are able to preserve high-frequency content and yield crisper interpolated images [88, 89].

- The proposal to shift visual effects design to the post-production stage, Ch. 8. We show that the Virtual Video Camera [94] discussed in Ch. 3 is able to synthesize any image sample required for visual effects design [90]. We further show in Ch. 9 that, using this system, not only visually plausible but also geometrically valid in-between views can be synthesized for stereoscopic content creation [79].

1.3 Outline of the Thesis

The focus of this work is on the presentation of a purely image-based end-to-end system for spatio-temporal image interpolation, the development of methods for the estimation of high-quality, long-range dense correspondence fields for the task of image interpolation, and on algorithms for the interpolation of novel viewpoints from sparse input footage. In the next chapter, we introduce basic concepts. Afterwards, we discuss the

Virtual Video Camera, a purely image-based free-viewpoint navigation system, which forms the framework for the algorithms developed within this thesis, Ch. 3. We then investigate the use of multi-exposure images for dense optical flow computation. We show that under controlled acquisition setups, this approach can be used to capture fast motion and to derive dense flow fields for fast action, Ch. 4. In the following chapter, an analysis and evaluation of current optical flow algorithms is presented for the task of image interpolation. We show that despite the subpixel accuracy of current optical flow algorithms, their performance with respect to perceived quality still leaves room for improvement, Ch. 5. Drawing conclusion from the evaluation, we then present a symmetric, long-range formulation for optical flow estimation based on SIFT descriptors, Ch. 6. We show that symmetric flow fields lead to improved image interpolation results, and that SIFT descriptors increase the robustness of the estimation over long ranges, varying lighting conditions and complex scenes. In Ch. 7, we exploit the symmetric flow fields derived in Ch. 6 to formulate multi-image interpolation as a labeling problem. We show that by having only one image source per pixel, ghosting and blurring artifacts common to blending-based interpolation algorithms can be completely avoided. Putting the algorithms to work within the Virtual Video Camera, we show how this system can be used to create high-quality visual effects from sparse, uncalibrated footage in a post-processing stage in Ch. 8 and use it to create stereoscopic content from the same footage in Ch. 9. We summarize in Ch. 10 and give an outlook on future research directions.

2

Background

2.1 The Plenoptic Function

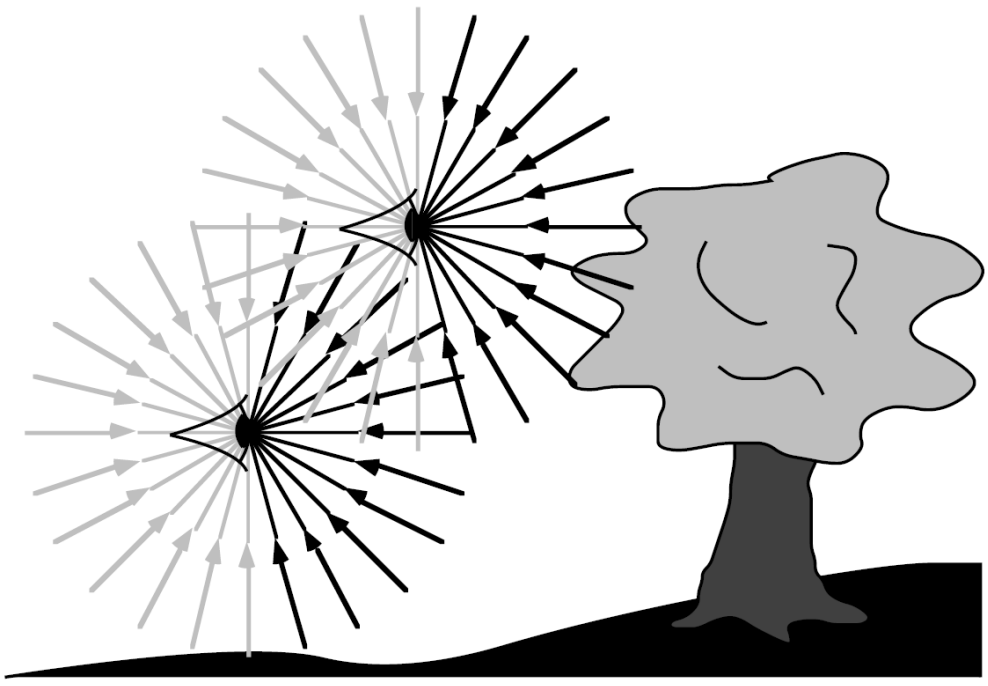


Figure 2.1: The plenoptic function describes the flow of light through a particular point in space and time. Image courtesy of Adelson and Bergen [1].

People see the world as an interaction of light with the surfaces of the objects surrounding them. The *plenoptic function* [1] is a geometrical optics view of the flow of light in the world. In geometrical optics, light is assumed to travel in straight lines

in vacuum and in air. Light is emitted by every object in our world, travels straight out in every direction and interacts with other objects by refraction, reflection and absorption. The interaction with objects changes the properties of the light rays, e.g. the direction or its wavelength. Putting an observer in an arbitrary location in space, there is a collection of light rays going through that particular location. Formalizing this concept into an equation, it can be described as a 7-dimensional function

$$P(\theta, \phi, \lambda, t, O_x, O_y, O_z),$$

where θ and ϕ describe the viewing angle, λ denotes the wavelength, t the time instant and O_x, O_y, O_z describe the position of the observer in space. The plenoptic function hence describes the time-dependent radiance contributed by a certain wavelength for a light ray. Clearly, only a small subset of this function can be measured by an observer: images taken with a camera or the retinal image of the eye are just incomplete samples of this function for a particular point in space O_x, O_y, O_z and time instant t.

In this thesis, we focus on methods based on sparse incomplete samples of the plenoptic functions such as images and video sequences. Our goal is to reconstruct parts of the plenoptic function from the measured sparse set of incomplete samples and to interactively navigate on the hull spanned by the plenoptic samples.

2.2 Image Morphing

Image morphing is a technique in computer graphics that transforms one image into another through a seamless transition. Its is nowadays a common special effect in movie picture production. This section summarizes the introduction given in Ref. [138].

The plausibility and quality of image morphing depends on the quality of the two major processes involved: image warping of the images and the following blending of the warped input images. Image warping describes the geometric deformation of the image lattice to bring corresponding features into alignment, image blending then generates the smooth transition between the two images by per-pixel blending of color values. The in-between image $I_{1,2}(\alpha)$ for $\alpha \in [0 \dots 1]$ is hence defined as

$$I_{1,2}(\alpha) = B(W(I_1, \alpha), W(I_2, 1-\alpha), \alpha) \tag{2.1}$$

with the boundary conditions $I_{1,2}(0) = I_1$ and $I_{1,2}(1) = I_2$. W denotes the geometric deformation, or warping function, and B is the blending function. A smooth transition

between image pairs can be achieved by computing a series of in-between images $I_{1,2}(\alpha)$ for regularly spaced α. As blending function B, most image morphing techniques employ a simple linear cross-dissolve. This hints at the fact that the warping step has a far greater influences on the plausibility of the transformation than the blending step. Especially the alignment of geometric features such as lines, edges or corners play an important role for visually plausible transitions. This is also in accordance with the knowledge we have about the human visual system which focuses on edges [75, 102].

2.2.1 Image Warping

The major differentiating factor for image morphing techniques is hence the spatial transformation used as warping function W. A spatial transformation in this context denotes a - not necessarily invertible - mapping between two coordinate systems that establishes correspondences between an image and its warped counterpart.

Put into an equation, a spatial transformation relates source coordinates x_1 to target coordinates x_2 or vice versa, i.e.

$$x_2 = W_f(x_1) \tag{2.2}$$

and

$$x_1 = W_b(x_2). \tag{2.3}$$

The forward warping approach, Eq. (2.2) typically leads to sampling artifacts: a single pixel in the target image may receive contributions from several source pixels or may receive no contribution at all, resulting in holes. To overcome this problem, forward warping is typically implemented by splatting the source pixel to the target domain. Alternatively, it can also be implemented on modern programmable graphics hardware by deforming 2D meshes with one quad per pixel in a vertex shader.

Backward mapping circumvents the problem of holes in the image by performing an inverse lookup of the target coordinate in the source image. However, a prerequisite for this is that the inverse transformation exists and can be computed. While this holds true for affine or perspective transformations, this property does not hold in the general case. If it exists, backward warping assures that each pixel in the target receives a contribution, and no holes or overlaps will occur. Backward warping can be implemented on modern graphics hardware by per-pixel texture lookup in the fragment shader with appropriate texture filtering operations enabled to counter aliasing artifacts.

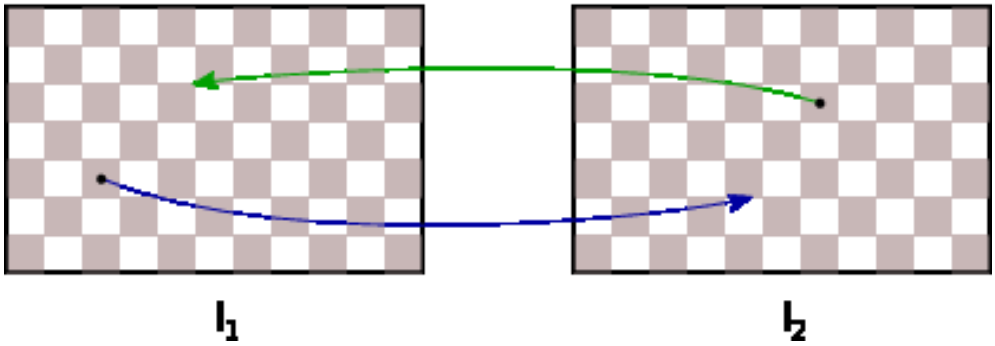


Figure 2.2: Forward vs. backward warping. Forward warping (blue) is the mapping of input integer lattice coordinates to target coordinates not necessarily on the integer lattice. Backward warping (green) is the opposite direction and uses the inverse transformation to map the target coordinates back to the integer lattice.

A variety of spatial transformations have been used for image morphing: global transformations such as affine or perspective transformations, piecewise spatial transformations which are restricted to disjoint regions of the image lattice, or in the most general case per-pixel transformations such as optical flow fields. We will not go into further detail here but refer the interested reader to the book of Hartley and Zisserman [68] for a thorough discussion of spatial transformations in general and to the report of Wolberg [167] for a detailed discussion of warping approaches. For the scope of this thesis, we employ forward warping with general optical flow fields as deformation function in the image morphing.

2.2.2 Image Blending

Image compositing techniques combine two or more images into a single result. In the context of image morphing, image blending interpolates the appearance of the input images after their geometric shapes have been brought into alignment. Each input image is weighted by the distance ratio of the current in-between image α, Eq. (2.1). The simplest blending scheme often employed for image morphing is known as cross-dissolve,

$$B(I_1, I_2, \alpha) = (1 - \alpha)I_1 + \alpha I_2.$$

Despite its simplicity, this approach often yields high quality results that do not need further improvement. However, there are two problems where a spatially-variant and

non-linear image blending is advantageous. The first type of artifacts arising during interpolation is due to misalignment of corresponding features which manifests itself as ghosting. The second is to account for regions that are visible in one of the images but not in the second one. When interpolating between images where parts are occluded, then some information is missing in one of the images and thus creates artifacts when blended linearly. This can be countered by non-linear adaptive blending approaches as presented by Grundland et al. [65]. However, even with such non-linear adaptive approaches, blending remains a low-pass filtering operation. In this thesis, we investigate an approach that avoids blending completely and thus preserves high-frequency content, Ch. 7. The problem with disoccluded regions is tackled within our label-based approach by identifying and discarding those regions during the creation of the interpolated image.

2.3 Optical Flow

Following Horn's taxonomy [73], the *optical flow* is the *apparent motion* of the brightness pattern in the image. This is in contrast to the *motion field* which is defined as the 2D projection of the real 3D motion of surfaces in the world. These two are not always the same and, in practice, the goal of optical flow recovery is application-dependent. In image morphing, it may be preferable to estimate the apparent motion so that, for example, highlights move in a realistic way.

Given two images $I(x, y, t)$ and $I(x, y, t + \delta t)$, the apparent motion is computed using the assumption of *brightness constancy*, that is, pixel intensities of corresponding 3D scene points are assumed to be the same in both 2D projections:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t). \tag{2.4}$$

Interestingly, brightness constancy is frequently violated when the scene contains shadows, specular highlights, or occluding objects, since occlusion is often not modelled explicitly. Further on, sensor noise can lead to a violation of the assumed brightness constancy.

Applying a Taylor series expansion to Eq. (2.4) and ignoring higher order terms, one gets

$$I(x + \delta x, y + \delta y, t + \delta t) \approx I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t. \tag{2.5}$$

The terms $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial t}$ are partial image derivatives. Inserting Eq. (2.5) into Eq. (2.4) results in the gradient constraint equation

$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} = 0. \tag{2.6}$$

Eq. (2.6) provides one equation in two unknowns δx and δy per pixel but only imposes one local constraint on image motion. Only the motion component in the direction of the local gradient of the image intensity can be estimated. This phenomenon is known as the aperture problem. For example, the motion of a homogeneous region cannot be recovered optically since the image gradient does not provide any information.

Additional constraints are hence necessary to obtain a unique solution. Frequently, the assumption is made that the motion field changes smoothly between neighboring pixels. This is known as the classical Horn-Schunck approach [73] or the locally constant Lucas-Kanade approach [98].

The differential approximation made in Eq. (2.6) is only valid for small displacements of at most one pixel. A common solution to allow for estimation of larger displacements is to use a multi-resolution coarse-to-fine approach: the optical flow is computed on the coarsest resolution, and an upsampled version of δx and δy is used to initialize the solution on the next finer level. This process is then iterated until the final image resolution is reached. However, as soon as the motion of the object becomes larger than the object itself, the motion cannot be faithfully recovered since the object vanishes at some resolution in the coarse-to-fine approach.

Optical flow research in the last 5 years concentrated mostly on different data terms, e.g. gradient-based data terms or color differences in different color spaces, on different regularization strategies, e.g. quadratic, robust or total variation regularizers, and different optimization strategies. Recently, there is a trend towards long-range motion estimation techniques, specifically tailored to large pixel displacements. Based on these approaches, we develop a robust long-range optical flow algorithm tailored to the needs of image morphing in Ch. 6.

2.4 Free Viewpoint Video

Free-viewpoint video (FVV) denotes a new form of visual medium that has received considerable attention in the last years [35, 43, 100, 104, 105, 135, 152, 181]. The

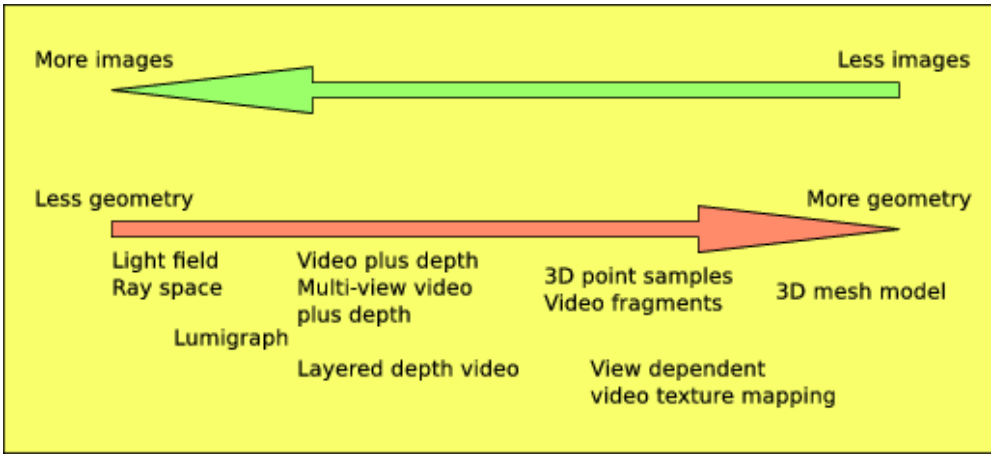


Figure 2.3: Spectrum of IBR representations

user is offered the possibility to move freely around the scene and to take on every possible viewpoint. Its goal is the interactive display of real dynamic scenes, captured by cameras. Common to all FVV systems is that they are based on a set of incomplete samples of the plenoptic function, e.g. multi-view recordings captured from different viewpoints and viewing angles. FVV systems differ in the format they use for 3D scene representation. There are two extreme representations: geometry/model-based and image-based, with a large "continuum" of representations with varying amount of geometry in between, Fig. 2.3. No free-viewpoint navigation system is yet able to offer a unifying solution allowing navigation of a scene in space as well as time in high-quality while minimizing the number of restrictions on the type of scene and acquisition effort.

Geometry or model-based FVV systems are either based on a 3D reconstruction of the scene or they employ a generic model of the scene, e.g. to model an actor in the foreground [35, 43]. Such systems can offer high-quality view interpolation and have the appealing property of unrestricted choice of camera position and viewing direction. However, because they require scene reconstruction, they are limited to mostly Lambertian scenes, and the footage has to be captured with synchronized hardware in a controlled environment. In addition, geometry and model-based systems are frequently only considering restricted scene foreground. The objects of interest have to be extracted before reconstruction, either by manual interaction or by error-prone image processing algorithms. While view interpolation comes nearly for free, temporal interpolation additionally requires the reconstruction of the scene motion. These restrictions

have so far prevented a wide-spread use of geometry-based FVV systems.

Purely image-based FVV systems have been around as well for 15 years [64, 86]. Being based on images alone, they have the potential for photo-realistic visualization. Further on, in contrast to geometry-based systems, they are not restricted to Lambertian scenes. However, these advantages come at the cost of requiring dense sampling of the plenoptic function which is needed for plenoptic sample interpolation. Similar to geometry-based approaches, these methods often also rely on synchronously captured footage.

In this thesis, we concentrate on algorithms for purely image-based FVV systems. The distinction of the system used in this thesis, Ch. 3, to previous systems is that it operates on a sparse set of unsynchronously captured plenoptic samples. The system treats the spatial domain on par with the temporal domain and can hence simultaneously interpolate in space as well as time. In order to be able to reconstruct plenoptic samples, it employs image morphing with general per-pixel correspondence fields used for image deformation.

2.5 Visual Effects Design

Visual effects are nowadays a frequent ingredient in motion pictures and TV commercials. They can be broadly categorized by way of production: many effects are based on traditional 3D computer graphics, others, such as space-time visual effects, are image-based and are created on-set with specific multi-camera setups [48]. The most popular space-time effect is "Bullet Time" known from the motion picture "The Matrix" [8]. In this effect, scene motion seems to be frozen while the camera moves around the scene. To create this effect, hundreds of cameras were placed along the desired final camera trajectory and had to be triggered simultaneously, Fig. 2.4. This corresponds to a dense sampling of the plenoptic function along the intended camera path. The individual cameras' stills are then concatenated to form the final video sequence. While such an approach has the advantage that each individual frame has been recorded directly, the camera path is fixed and cannot be altered afterwards.

Other space-time visual effects are created in a similar way and also involve specific camera setups. To methodically describe all possible effects and to aid in the design of respective camera setups, Wolf introduced an intuitive graphical notation termed

Figure 2.4: Camera setup used for the famous Bullet Time from the motion picture "The Matrix" [8]

space-time diagrams [168]. In such a diagram one axis encodes the spatial, the other the temporal dimension and the arrangement of the frames within this space-time plane completely defines the visual effect. This notation is used extensively by the company Digital Air to design many intriguing effects [48].

In this thesis, we adapt the graphical notation of the space-time diagrams of Wolf and extend it to three dimensions to form an intuitive special effects navigation space. Based on the Virtual Video Camera system, various visual effects can be designed and synthesized from a sparse set of plenoptic samples in post-production, without the need to meticulously plan each effect in advance and to configure camera positions.

3

The Virtual Video Camera

3.1 Introduction

This chapter presents the Virtual Video Camera system which forms the framework for the algorithms proposed in this thesis. The Virtual Video Camera is a purely image-based free-viewpoint navigation system which features a continuous navigation in space and time [94]. In this chapter, we describe the full end-to-end pipeline of the system from acquisition to rendering.

The objective common to all free-viewpoint navigation systems is to render photo-realistic vistas of real-world, dynamic scenes from arbitrary perspective (within some specified range), given a number of simultaneously recorded video streams. To date, most systems exploit epipolar geometry based on either dense depth/disparity maps [63, 114, 181] or complete 3D geometry models [35, 43, 103, 104, 135, 152]. In order to estimate depth or reconstruct 3D geometry of dynamic scenes, the input multi-video data must be precisely calibrated as well as captured synchronously. This dependence on synchronized footage can limit practical applicability: high-end or even custom-built acquisition hardware must be employed, and the recording setup indispensably includes some sort of camera interconnections (cables, WiFi). The cost, time, and effort involved in recording synchronized multi-video data prevents widespread use of free-viewpoint navigation methods. Further, most existing systems are designed to interpolate virtual camera positions only along spatial dimensions. Temporal view interpolation requires additional scene motion information [152, 153].

The Virtual Video Camera system proposed by Lipski et al. addresses these limitations [94]. This approach accepts unsynchronized, uncalibrated multi-video footage as input. It is motivated by the pioneering work on view interpolation by Chen and Williams [38]. They pick up on the idea of interpolating different image acquisition attributes in higher dimensional space and suitably extend it to be applicable to view interpolation in the spatial as well as temporal domain. By putting the temporal dimension on a par with the spatial dimensions, a uniform framework is available to continuously interpolate virtual video camera positions across space and time. Instead of using depth/disparity or 3D geometry, the system makes use of dense image correspondences and is thus applicable to scenes whose object surfaces are highly variable in appearance, e.g. due to specular highlights, or hard to reconstruct for other reasons. Perceptually plausible image correspondence fields can often still be established where ground-truth geometry (or geometry-based correspondences) cannot [17]. Dense image correspondences can also be established along the temporal dimension to enable interpolation in time.

The rest of this chapter is structured as follows: we first highlight related free-viewpoint navigation systems in the next section. We then discuss the navigation space of the Virtual Video Camera in detail, Ch. 3.3, for it will be required in later chapters in this thesis. We briefly touch upon navigation space tessellation in Ch. 3.4 before discussing the remaining parts of the full end-to-end pipeline of the Virtual Video Camera, Fig. 3.1, in Ch. 3.5. We briefly summarize benefits and limitations of the system in Ch. 3.6.

3.2 Related Work

In general, free-viewpoint video systems can be classified by the amount of geometry used to synthesize novel views. This classification has been widely used in literature and we follow the classification given in Stich's PhD thesis [138] and extend it suitably.

3.2.1 Image-based FVV.

Light fields, lumigraph and ray space. Image-based reconstruction of the plenoptic function has first been introduced by Levoy and Hanrahan [86] as light field rendering. In their approach, Levoy and Hanrahan densely sampled the static scene with

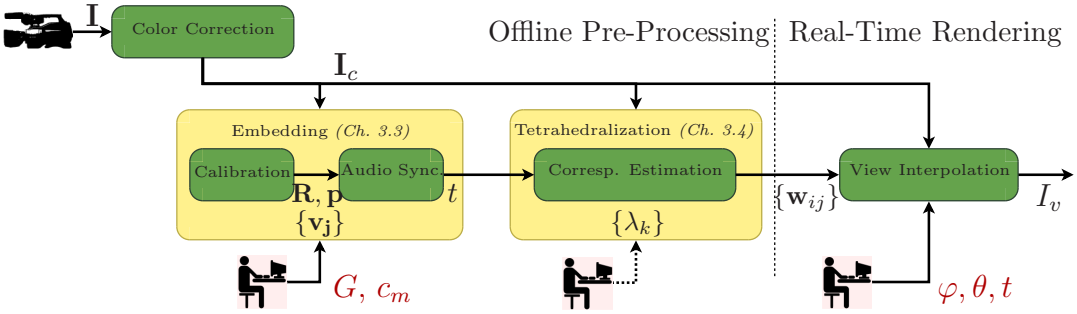


Figure 3.1: Virtual Video Camera processing pipeline: multi-video data ($\mathbf{I}$) is color-corrected first ($\mathbf{I}_c$). To embed the video frames into navigation space $\{\mathbf{v_j}\}$, the user specifies the master camera c_m and a common ground plane G . Extrinsic camera parameters $(\mathbf{R}, \mathbf{p})$ and the time offsets (t) are automatically estimated. Adjacency of video frames induces a tetrahedralization $(\{\lambda_k\})$ of the navigation space, and dense correspondences $(\{\mathbf{w}_{ij}\})$ are estimated along the edges of the tetrahedralization. After these off-line processing steps, the navigation space can be interactively explored (viewing directions φ, θ, time t) by real-time rendering the virtual view I_v.

cameras placed on a regular grid. A novel view is synthesized by interpolating a 4D function. At the same time, Gortler et al. [64] proposed the lumigraph rendering which is similar in spirit to light field rendering but can handle unstructured data. They re-sample the input data into the lumigraph data structure and additionally make use of a proxy geometry to improve rendering results. In 2001, Buehler et al. [33] introduced the unstructured lumigraph which renders the additional re-sampling step of Gortler et al. obsolete. Another approach based on the reconstruction of the plenoptic function was proposed by McMillan and Bishop [108]. Their system interpolates a novel view based on cylindrical projections of a scene taken at different spatial locations. While these approaches only showed static scenes due to synchronized capture, they can be readily applied to dynamic scenes as well: Fujii and Tanimoto [56] present a FVV system based on a ray-space representation of the captured scene. Further, Matusik and Pfister [105] proposed an auto-stereoscopic free-viewpoint system that relies on light field rendering. However, a major drawback of light field rendering is the dependence on synchronized dense data acquisition, requiring specialized hardware setups [163, 164, 174] and leading to a large amount of data.

The dependence on synchronized recordings can be overcome by space-time light field rendering [153, 154]. The system operates on unsynchronized video streams and

is capable of interpolating in space and time. The input images are warped twice, first to a common virtual time before unstructured lumigraph rendering [33] is applied in a second step.

Warping-based. If additional depth information is provided, Schirmacher et al. [127] showed how to reduce the number of necessary input images while simultaneously improving rendering quality of the unstructured lumigraph approach. In contrast to explicitly reconstructing dense depth maps to improve the interpolation quality, Seitz and Dyer [130] determine the fundamental matrix to estimate dense disparity and warp-interpolate between two views of a static scene. Manning and Dyer [100] extended this approach to dynamic scenes by segmenting the scene into different motion layers and restricting the motions to rigid-body translations. Again restricted to rigid-body objects, Xiao and Shah [171] describe an interpolation method based on three input images. While those approaches make use of depth or disparity, Einarsson et al. [50] propose a purely image-based interpolation technique to synthesize plenoptic samples from sparse video footage which are then used in a light-field rendering approach. Stich et al. [139] propose a forward-warping approach that uses simple depth heuristics to resolve ambiguities in fold-over regions. In their approach, general 2D correspondence fields take the role of depth/disparity. Thus, it can also be applied to unsynchronized footage. Making further assumptions on the scene, e.g. assuming a well-defined background such as in a soccer stadium, Germann et al. [57] use a collection of articulated billboards to represent foreground objects. Their free-viewpoint system uses the a-priori knowledge on the background to extract foreground objects. Similar, Ballan et al. [13] present an image-based view interpolation that uses billboards to represent a single moving foreground actor. The background, however, is completely reconstructed using wide-baseline stereo techniques.

3.2.2 Model-based FVV

For the special domain of architecture Debevec et al. [44] introduced a method based on a coarse geometric model. The handcrafted geometry proxy is enhanced with a dense stereo reconstruction acquired from multi-view footage. Novel views can be rendered using view-dependent texturing. Carranza et al. [35] presented a free-viewpoint video system specialized to capturing the performance of a human actor. In their approach,

they recover pose parameters of a generic 3D model of the human body by fitting it to silhouette information extracted from multi-view video streams. Novel views of the actor are synthesized by rendering the 3D model with the recovered pose parameters and applying projective texturing to recreate the actors appearance. A drawback of this approach is that the human mesh model is generic and thus only imperfectly approximates the actor of interest. Recently, de Aguiar et al. [43] extended this approach by first scanning the actor in a static pose and then estimating the deformation of the mesh using a similar camera setup. With their approach they achieve high-resolution dynamic meshes of actors with arbitrary clothing.

3.2.3 Reconstruction-based FVV

If additional information about scene depth is available for each image pixel, this information can also be used to create in-between images. This has first been used by Chen and Williams [38] in their seminal work on view interpolation. Having the complete 3D geometry of a scene but being unable to display it in real-time, they proposed to use view morphing techniques to achieve interactive viewpoint navigation. Mark et al. [101] also followed this approach but also handled occlusion and discontinuities during interpolation rendering. As for real world scenes, no depth information is available from standard cameras, Zitnick et al. [181] reconstructed this information using a stereo approach. Based on dense depth maps, the system requires synchronized and calibrated multi-video footage which is acquired using a custom-built multi-camera system. With their reconstruction they were able to create high-quality view interpolations between a set of synchronized video cameras in real-time.

Instead of reconstructing the whole scene, several approaches restrict themselves to a single foreground object which is then inserted into a virtual environment. The visual hull approach of Matusik et al. [104] reconstructs a geometric proxy by intersecting silhouette cones extracted from a handful calibrated video streams. However, relying on correct foreground segmentation, the geometry reconstruction often suffers from cutting off small scale features such as fingers, or from filling in small gaps e.g. between arms and the body. This approach has been further refined by Matsuyama et al. [103] who proposed to reconstruct an initial model based on silhouettes and then deform the obtained mesh to faithfully capture dynamic aspects of the reconstruction. A different approach only considering foreground objects has been proposed by Würmlin et al.

[169]. They propose to represent the scene by images augmented with a depth layer; novel viewpoints are rendered by re-projecting each pixel into 3D space and splatting the point into the output view. Waschbüsch et al. [156] also based their view synthesis on a surfel model, however, they are not restricted to the foreground. To improve the quality of 3D reconstruction, they use a structured-light approach to obtain per-view depth maps. Hornung and Kobbelt [74] reconstruct a particle point cloud from an unordered image collection and use this representation for their rendering.

All approaches so far do not consider temporal coherence during reconstruction. This has been introduced by Goldlücke and Magnor [61, 62] who reconstruct a 4D space-time surface. Recently, Starck and Hilton [135] have also introduced improvements on the geometric reconstruction to achieve high quality and high resolution geometries from sparse synchronized camera setups based on silhouette segmentation. While these methods improved on the quality of the reconstructed geometry, the quality of the final free-viewpoint video is also strongly dependent on the appearance or texture of the model. Especially insufficient camera calibration accuracy and remaining deviations from the true 3D surface lead to artifacts such as ghosting and wrong textures at occlusion boundaries. Eisemann et al. [52] proposed a method based on warping of the camera images before projection to correct these problems. Their method can be performed in real-time on recent graphics hardware and thus instantly improves rendering results.

While most systems perform only spatial viewpoint interpolation, Vedula et al. [152] describe a volume-based approach that is capable of interpolating in space and time by estimating 3D geometry and motion. Klose et al. [80] recently designed a scene flow reconstruction that is able to cope with unsynchronized multi-view recordings.

The Virtual Video Camera system employed in this thesis is purely image-based. Lipski et al. pick up on the idea of Chen and Williams [38] of image interpolation in a higher-dimensional space of acquisition attributes and suitably extend it to view interpolation in the spatial and temporal domain. Being based on dense image correspondences, this system can operate on unsynchronized multi-view footage, and is able to synthesize novel views in space as well as time.

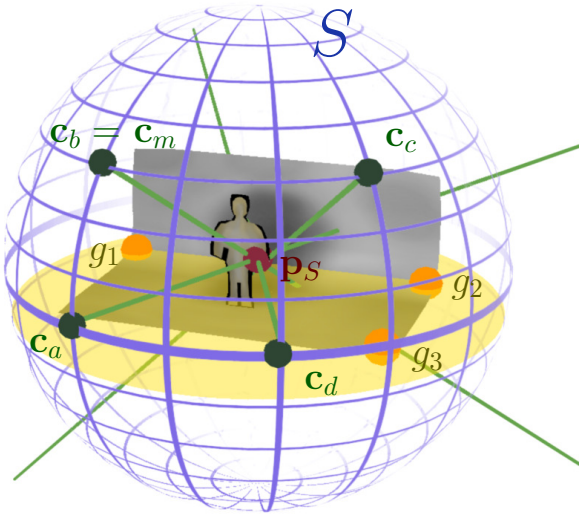


Figure 3.2: Navigation space: [94] define a sphere S around the scene. For the center $\mathbf{p}_S$ of S, they least-squares-determine the point closest to the optical axes of all cameras (green lines). The user selects three points g_1, g_2, g_3 to define the ground plane (yellow circle). They take the normal of the plane as the up vector of the scene, and thus as the rotation axis of the sphere. The embedding is uniquely defined by labeling one of the cameras as the Master camera $\mathbf{c}_m$.

3.3 Navigation space embedding

The goal of the Virtual Video Camera system is to explore a captured scene in an intuitive way and render a (virtual) view I_v of it, corresponding to a combination of viewing direction and time. To this end, Lipski et al. chose to define a 3-dimensional navigation space $\mathcal{N}$ that represents spatial camera coordinates as well as the temporal dimension. In their seminal paper, Chen and Williams [38] propose to interpolate the camera rotation $\mathbf{R}$ and position $\mathbf{p}$ directly in 6-dimensional hyperspace. While this is perfectly feasible in theory, Lipski et al. show that it has several major drawbacks in practice: it neither allows for intuitive exploration of the scene by a user, nor is it practical to handle the amount of emerging data needed for interpolation in this high-dimensional space. Additionally, cameras would have to be arranged in a way that they span an actual volume in Euclidean space. With this requirement, it would be hard to devise an arrangement of cameras where they do not occlude each other's view of the scene. The crucial design decision in the Virtual Video Camera system

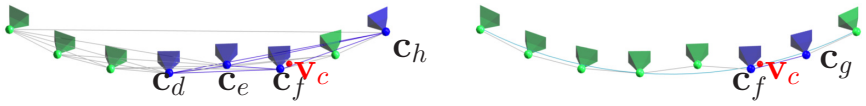


Figure 3.3: Two possibilities to partition the camera setup of the *Breakdancer* sequence. Both images depict the camera setup seen from above. When partitioning the Euclidean space directly several problems arise. Although an actual volume is spanned by the cameras, many tetrahedra are degenerated. If the virtual camera $\mathbf{v}_c$ is reconstructed, captured images from cameras $\mathbf{c}_d$, $\mathbf{c}_e$, $\mathbf{c}_f$ and $\mathbf{c}_h$ as well as the wide-baseline correspondence fields between them are required. In the Virtual Video Camera navigation space embedding (right), interpolation only takes place between neighboring cameras, less data has to be processed and correspondence estimation is much easier. Additionally, spatial navigation is intuitively simplified. Instead of a 3D position, only the position on the one-dimensional arc (turquoise) has to be specified. The small spatial error (distance between $\mathbf{v}_c$ and line segment between $\mathbf{c}_f$ and $\mathbf{c}_g$) is negligible even for loose setups.

is hence to map the extrinsic camera parameters to a lower dimensional space that allows intuitive navigation. The temporal dimension already defines one axis of the navigation space $\mathcal{N}$, leaving two dimensions for parameterizing the camera orientation and position. Practical parameterizations that allow for realistic view interpolation are only possible if the cameras' optical axes cross at some point (possibly at infinity).

A natural choice for such an embedding is a spherical parameterization of the camera setup. While for example a cylindrical embedding or an embedding in a plane is also feasible, a spherical embedding allows for all reasonable physical camera setups, ranging from cameras arranged in a 1-dimensional arc, over cameras placed in a spherical setup to linear camera arrays with parallel optical axes in the limit. Regarding existing data sets, it is obvious that spherical/arc-shaped setups are the predominant multi-view capture scenarios (e.g. [43, 181]). Other placements, such as panoramic views, are also possible, but would suffer from the small overlap of the image regions of cameras.

Cameras are placed on the surface of a virtual sphere, their orientations are defined by azimuth φ and elevation θ. Together with the temporal dimension t, φ and θ span a 3-dimensional navigation space $\mathcal{N}$. In contrast to the conventional partition of space suggested by Chen and Williams [38], Lipski et al. thus restrict the movement of the virtual camera to a subspace of lesser dimensionality (2D approximate spherical surface or 1D approximated arc). Although this might appear as a drawback at first sight, several advantages arise from this crucial design decision, see Fig. 3.3:

1. The amount of correspondence fields needed for image interpolation is reduced significantly, making both pre-processing and rendering faster.

2. An unrestricted partition of Euclidean space leads to degenerated tetrahedra. Especially when cameras are arranged along a line or arc, adjacencies between remote cameras are established for which no reliable correspondence information can be obtained.

3. The parameterization of the camera arrangement provides an intuitive navigation around the scene.

To define the navigation space $\mathcal{N}$, Lipski et al. assume that they know ground-truth extrinsic camera parameters $\mathbf{R}$ and $\mathbf{p}$ for every camera, as well as a few point correspondences with their 3D world coordinates. For a specific virtual image I_v, they want to interpolate the image at a given point in navigation space defined by the two spatial parameters φ and θ as well as recording time t . To serve as sampling points, the camera configuration of a recorded multi-video input in Euclidean world space is embedded into navigation space $\mathcal{N}$

$$\Psi : (\mathbf{R}, \mathbf{p}, t) \mapsto (\varphi, \theta, t).$$

In the Virtual Video Camera system, Ψ is simply a transformation from Cartesian coordinates to spherical coordinates, where the sphere center $\mathbf{p}_S$ and the radius of the sphere r_S are computed from the cameras' extrinsic parameters $\mathbf{R}$ and $\mathbf{p}$ in a least-squares sense. The embedding is uniquely defined by specifying a ground plane in the scene and by labeling one of the cameras as the master camera $\mathbf{c}_m$, cf. Fig. 3.2.

3.4 Space-time tetrahedralization

The embedding Ψ results in a three-dimensional point cloud, each point (φ, θ, t) representing a sample of the plenoptic function. In order to reconstruct an arbitrary virtual view from this sample set, a partition of the 3D space is needed to efficiently fetch the closest samples. For a d-dimensional space, Chen and Williams proposed to generate some arbitrary graph to partition the space such that every possible viewpoint lies in a d-simplex and can be expressed as a linear combination of the $d + 1$ vertices of the enclosing simplex [38]. Lipski et al. extend this idea and apply a constrained Delaunay

tetrahedralization to the set of navigation space points. The tessellation is constrained such that every tetrahedron consists of at most three vertices with different φ and θ, i.e. three different real cameras. The fourth vertex always represent an image of one of the other three cameras shifted along the temporal axis. This is necessary to avoid artifacts during rendering since cameras only map approximately to the sphere surface due to the least-squares approach. For a detailed discussion of this topic, we refer the reader to Ref. [94].

3.5 Processing Pipeline

Based on the navigation space tetrahedralization described above, Lipski et al. propose a fully functional processing pipeline for free-viewpoint navigation from unsynchronized multi-video footage. Their processing pipeline makes use of known techniques and suitably adapts them to solve the encountered problems.

3.5.1 Acquisition

To acquire multi-video data, they use up to 16 HDV Canon XHA1 camcorders (1440 x 1080 pixels, 25 fps). The captured sequences are internally MPEG-compressed, stored on DV tape, and later transferred to a standard PC. This setup is very flexible, easy to setup, runs completely on batteries and is suitable for indoor and outdoor use. Static setups using tripods as well as setups with dynamic hand-held cameras are possible. Adjacent cameras should only have sufficient view overlap to facilitate correspondence estimation. For enhancing interpolation, the angle between neighboring cameras should not exceed roughly 10 degrees, in vertical or horizontal direction. Of course, the total range of possible virtual views is determined by camera configuration. Fig. 3.4 shows some typical camera configurations.

3.5.2 Pre-processing

Color correction. To correct for color balance differences among cameras, Lipski et al. use the master camera $\mathbf{c}_m$ defined in the embedding stage and apply the color correction approach presented by Snavely et al. to all video frames [134].

Figure 3.4: Camera setups: the Virtual Video Camera system uses consumer-grade camcorders, mounted either on tripods or handheld.

Camera calibration. They also need to determine $\mathbf{R}$ and $\mathbf{p}$ of the camera setup to define the mapping Ψ from world space coordinates to navigation space $\mathcal{N}$, Ch. 3.3. Recent structure-from-motion algorithms for unordered image collections [59, 93, 129] solve this problem robustly and can also provide a set of sparse world space points needed for constructing the common ground plane for our navigation space. Lipski et al. report that this algorithm yields robust results also for dynamic scenes. For dynamic camera scenarios (i.e., hand-held, moving cameras), $\mathbf{R}$ and $\mathbf{p}$ have to be computed for every frame of each camera.

Temporal registration. The mapping Ψ additionally needs the exact recording time t of each camera. Lipski et al. estimate the sub-frame temporal offset by recording a dual-tone sequence during acquisition and analyzing the audio tracks afterwards [69]. If recording a separate audio track is not feasible, pure post-processing approaches [110] can be employed instead.

Dense correspondence field estimation. In order to interpolate between two images I_i, I_j, bidirectional dense correspondence maps $\mathbf{w}_{ij}$ for each tetrahedral edge in navigation space $\mathcal{N}$, Ch. 3.3, are needed. In the Virtual Video Camera system, Lipski et al. employ the algorithm proposed by Stich et al. [140; 141]. For moderately complex scenes, the results are convincing, and if not, the algorithm accepts manual corrections.

3.5.3 Rendering

Having subdivided navigation space $\mathcal{N}$ into tetrahedra, each point $\mathbf{v}$ is defined by the vertices of the enclosing tetrahedron $\lambda = \{\mathbf{v}_i\}, i = 1 \dots 4$. Its position can be uniquely expressed as $\mathbf{v} = \sum_{i=1}^{4} \mu_i \mathbf{v}_i$, where μ_i are the barycentric coordinates of $\mathbf{v}$. Each of the 4 vertices $\mathbf{v}_i$ of the tetrahedron corresponds to a recorded image I_i. Each of the 12 edges e_{ij} correspond to a correspondence map $\mathbf{w}_{ij}$, that defines a translation of a pixel location $\mathbf{x}$ on the image plane. We are now able to synthesize a novel image I_v for every point $\mathbf{v}$ inside the recording hull of the navigation space $\mathcal{N}$ by multi-image interpolation:

$$I_v = \sum_{i=1}^{4} \mu_i \tilde{I}_i, \tag{3.1}$$

where

$$\tilde{I}_i \left(\mathbf{\Pi}_i \mathbf{x} + \sum_{j=1,\dots,4, j \neq i} \mu_j (\mathbf{\Pi}_j (\mathbf{x} + \mathbf{w}_{ij}(\mathbf{x})) - \mathbf{\Pi}_i \mathbf{x}) \right) = I_i(\mathbf{x}) \tag{3.2}$$

are the forward-warped images [101]. $\{\mathbf{\Pi}_i\}$ defines a set of re-projection matrices that map each image I_i onto the image plane of I_v, as proposed by Seitz and Dyer [130]. Given the calibration matrices $\mathbf{K}_i$ and $\mathbf{R}_i$ for each camera, the re-projection $\mathbf{\Pi}_i$ can be computed as

$$\mathbf{\Pi}_i = \mathbf{K}_v \mathbf{R}_v (\mathbf{K}_i \mathbf{R}_S \mathbf{R}_i)^{-1},$$

where

$$\mathbf{K}_v = \begin{pmatrix} f_i & 0 & \frac{w}{2} + o_x \\ 0 & f_i & \frac{h}{2} + o_y \\ 0 & 0 & 0 \end{pmatrix}$$

is the intrinsic matrix of the virtual view composed of the focal length f_i of the camera , the width w and height h of the virtual view in pixels and the center offsets o_x and o_y of the re-projection of the sphere center $\mathbf{p}_S$ into I_i. $\mathbf{R}_S$ defines the rotation of the spherical embedding introduced in Ch. 3.3 and $\mathbf{R}_v$ denotes the rotation matrix of the virtual view I_v. Since the virtual image I_v is always oriented towards the center of the scene, this re-projection corrects the skew of optical axes potentially introduced by the loose camera setup and also accounts for jittering introduced by dynamic cameras. Image re-projection is done on the GPU without image data re-sampling. Since the re-projection alters the source and target domain of the warp fields $\mathbf{w}_{ij}$, one has to account for this during the forward warping stage, Eq. (3.2), as illustrated in Fig. 3.5.

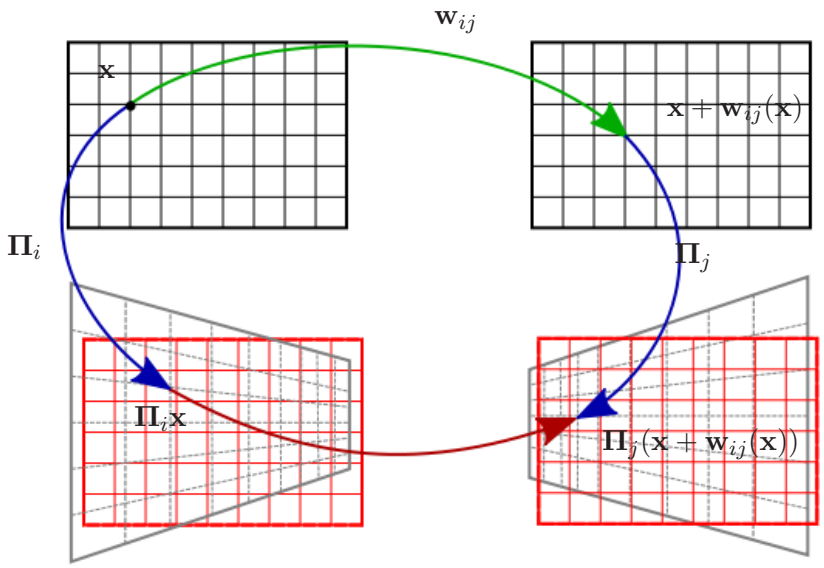


Figure 3.5: Image reprojection applied to image warping without re-sampling image data. Since image reprojection alters the source and target lattice used to compute dense correspondences fields $\mathbf{w}_{ij}$ in the preprocessing stage, this has to be compensated for during rendering. To this end, Lipski et al. first transform source $\mathbf{x}$ and target coordinates $\mathbf{x}+\mathbf{w}_{ij}(\mathbf{x})$ to the image plane of the virtual view using $\mathbf{\Pi}_i$ and $\mathbf{\Pi}_j$, respectively. The correct correspondence is then given by $\mathbf{\Pi}_j(\mathbf{x} + \mathbf{w}_{ij}(\mathbf{x})) - \mathbf{\Pi}_i\mathbf{x}$. This computation is carried out in a vertex shader during rendering.

Lipski et al. handle occlusion/disocclusion on-the-fly based on correspondence field heuristics as proposed by Stich et al. [139].

Disocclusions are detected by calculating local divergence in the correspondence fields. If any two neighboring pixels exhibit a difference of more than 4 pixels in their motion, their contribution to the interpolated images is suppressed by adjusting per-pixel blending weights accordingly. As a last step, the borders of the rendered images are cropped (10% of the image in each dimension), since no reliable correspondence information is available for these regions.

At 940×560 pixels output resolution, rendering frame rates exceed 25 fps on an NVIDIA GeForce 8800 GTX. Lipski et al. store images and correspondence fields in local memory and use a pre-fetching scheme for image data. This rendering approach is used for online rendering in their interactive viewer, Fig. 3.6. Within this viewer, the user can continuously change viewing direction and time by simple click-and-drag

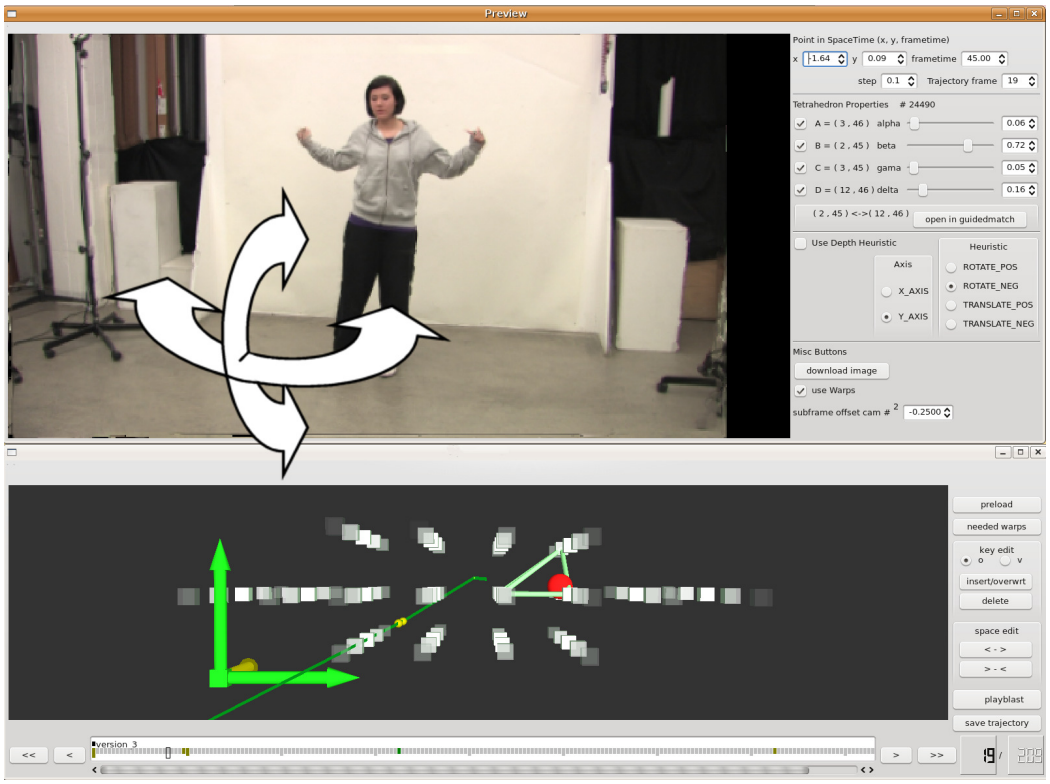


Figure 3.6: Virtual video editing: to interactively create arbitrary virtual video camera sequences, the user moves the camera by click-and-drag movements in the rendering window (top). The spline curve representing the space-time camera path is automatically updated and visualized in the navigation space window (bottom).

movements within the rendering window. Camera paths are defined by placing, editing, or deleting control points in navigation space at the bottom of the screen. The virtual video camera path through space-time is interpolated by Catmull-Rom splines [36].

3.6 Summary

The Virtual Video Camera system has been evaluated on a variety of real-world dynamic scenes and has been shown to deliver high-quality results [94]. Besides high-quality image interpolation in space and time from unsynchronized multi-view footage, the design of the navigation space allows intuitive exploration of the recorded space-time. As such, the Virtual Video Camera system lends itself as a valuable tool for intuitive design of visual effects and stereoscopic content creation as we will show in Ch. 8 and

Ch. 9 of this thesis.

However, rendering quality obviously depends on two aspects: the visual plausibility of the correspondence fields and the quality of the interpolation algorithm itself. The remaining chapters of this thesis contribute solutions to both aspects. While the automatic, pair-wise correspondence estimation algorithm by Stich et al. [140] yields convincing results on moderately complex scenes, it runs into problems if the structural complexity of the scenes increases. To this end, we evaluate current state-of-the-art in pair-wise correspondence estimation for use in multi-view interpolation algorithms, Ch. 5. We devise an algorithm for long-range correspondence estimation based on SIFT descriptors [97] that yields correct correspondences also for complex scenes where the approach by Stich et al. [140] fails, Ch. 6.

The view synthesis of the Virtual Video Camera is based on adaptively blending several input images. While this delivers good results if correspondences are perfect, ghosting artifacts appear if there is a slight misalignment. Further on, blending constitutes a low-pass filter and the virtual view will always look softer than the original. This thesis offers a solution to this by completely avoiding the blending of several images as we will show in Ch. 7.

4

Multi-Exposure Flow

4.1 Introduction

Photographic capture of high-speed motion fascinates artists and researchers alike. First investigations date back to 1878 when Eadweard Muybridge conducted his famous experiments to create serial images of a galloping horse. In the 1930's, Harold E. Edgerton perfected the use of stroboscope photography to create multi-exposure images of high-speed natural phenomena. Multi-exposure images collapse several consecutive images onto one common image plane. On that account, such images implicitly convey a lot of information about the ongoing motion. Most intriguingly, our human visual system is often able to guess the underlying motion pattern from such multiply exposed images.

Stroboscopic photography offers a way to visualize and analyze a wide range of high-speed phenomena. For example, it would be possible to record time-varying phenomena such as an explosion and to measure the trajectory of the particles over time. Potential applications of this technique include athletics to assist athletes in training, e.g., for a tennis player bringing his serve to perfection. Up to now, such application fields rely on specialized and expensive high-speed cameras. In this chapter, we intend to demonstrate that multi-exposure images can offer a low-cost alternative to specialized hardware. So far, multi-exposure imaging has been used predominantly in situations where it was possible to augment the captured scene with markers that can be robustly and individually identified [148]. In this chapter, we present a method for multi-exposure motion analysis which is completely non-invasive. Our method requires

Figure 4.1: Multi-exposure image of the author waving an arm.

no scene intrusion such as markers or other preparation. We only require the motion to be directed, continuous and smooth, i.e., the temporal ordering of the exposures has to map to a spatial ordering in the image plane. The input to our algorithm is the multi-exposure image to be analyzed, and a single-exposure image taken shortly before (or after) the multi-exposure image. This single-exposure is needed to initialize our iterative algorithm and to determine the direction of time. Our method is based on deformable shape matching, followed by a step to estimate a set of locally restricted aligning transformations. This set of transformations maps the initial exposure onto each exposure present in the multi-exposure image.

As main contribution of this chapter, we propose an algorithm for deducing dense motion vector fields from multi-exposure images. We incorporate Euclidean distance and shape context distance into a distance measure and use it for shape matching in cluttered environments. For increased robustness, we employ our algorithm in a multi-resolution framework. In addition, we propose a method to automatically find a set of piecewise local transformations that minimize the matching error for a given set of point correspondences.

The rest of this chapter is organized as follows. In Ch. 4.2, we review the current state-of-the-art. Chapter 4.3 discusses the representation of the scene in both single- and multi-exposure images used by our algorithm which we describe in Ch. 4.4. Finally, we present and discuss the results of the proposed algorithm for several scenes, Ch. 4.5.

4.2 Related Work

In general, automatically deducing motion vector fields from multi-exposed images is an ill-posed problem. Only for specific cases multi-exposure imaging has been used, e.g., to analyze the trajectory of a flying baseball and the hand posture of a pitcher [148]. In order to be able to analyze the motion, both the ball and prominent positions on the hand of the pitcher were augmented with markers that could be easily identified and tracked.

Shape recognition based on spatial configurations of a small number of key points, on the other hand, is a well-researched field. Belongie et al. [18] introduced the shape context descriptor which characterizes a particular point on the shape. In essence, it is a log-polar histogram of the relative coordinates of all other points. Similar points in two shapes will have a similar relative position in each shape and will ideally have a similar shape context. Shape context matching has been applied to a wide variety of object recognition tasks [18, 112] where background clutter is limited. Thayananthan et al. [147] propose an algorithm for shape matching based on shape contexts that is applicable also in cluttered environments. They propose to integrate figural continuity into the matching framework by imposing an ordering on the object contour and penalizing matches that violate that ordering.

A standard method for point registration based on Euclidean distance is the Iterated Closest Point (ICP) algorithm [22]. Correspondences are found based on inter-point distance. The transformation is estimated by minimizing the geometric error between point pairs. This algorithm is fast and converges to a local optimum. However, it requires good initial alignment of model and target shape. Another improved approach is the non-rigid point matching proposed in Refs. [40, 60] which is based on thin-plate spline interpolation [25]. In this work, the authors jointly estimate correspondences and a non-rigid transformation aligning the point sets. This approach has proven

convergence properties and can be extended to multiple transformations, given their spatial support is known a-priori.

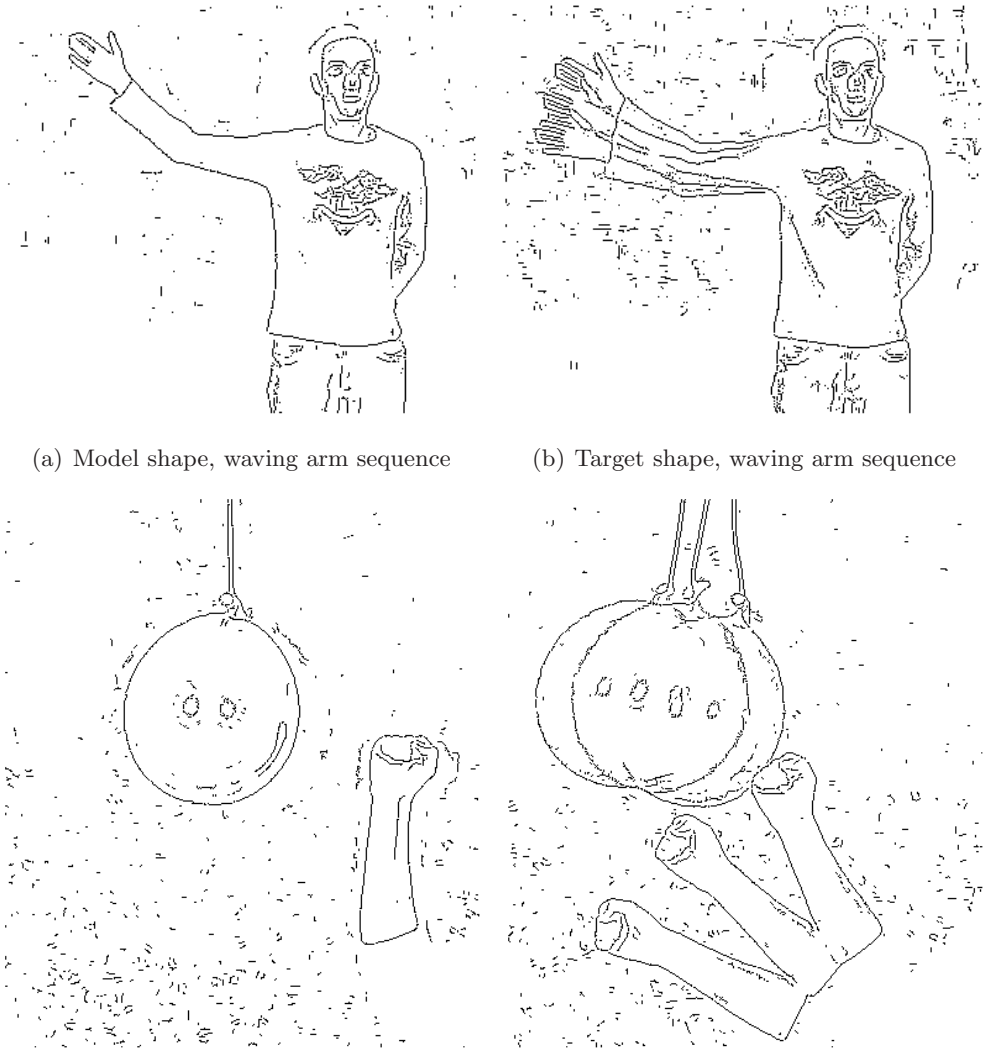
Concerning the estimation of dense deformation fields from a set of point correspondences, the most prominent work is the one of Bookstein [25]. In this work, the field is computed using thin-plate splines and radial basis functions for each point in the correspondence map. Recently, Schaefer et al. [126] proposed an algorithm to derive dense motion fields based on point correspondences and moving least squares interpolation. They incorporate affine, similarity and rigid transformations into a common framework. Both methods yield dense and globally smooth deformation fields.

Unfortunately, none of the mentioned shape-matching methods can directly be applied to multi-exposure images to find correspondences among overlapping scene parts. Euclidean distance-based methods such as ICP work well if the shapes are already coarsely aligned but produce wrong results if this is not the case. Shape contexts are the method of choice for matching only if the amount of clutter and distortion is expected to be small. In this chapter, we propose to combine the advantages of distance-based matching and shape context-based matching in a common framework, giving more weight to the Euclidean distance in already aligned areas and relying on shape contexts in areas where an alignment is not given. In addition, we propose a method for constructing discontinuity-preserving dense deformation fields from a set of sparse correspondences. This approach has also been applied in the context of perceptual image warping [139].

4.3 Shape representation

As input, we assume a multi-exposure image I_M containing k instances of an non-rigidly deforming object in motion, and a single-exposure image I_S of the same object before the motion sequence. Our task is to identify in I_M each of the k instances of the object depicted in I_S and compute aligning transformations that transform the single exposure image onto each of these instances. We assume linear superposition of the k instances in I_M, i.e., no saturation of pixel intensities due to multi-exposure.

Multi-exposure images are difficult to handle from an algorithmic point of view. Matching strategies based on color correlation or other intensity-based measures will



(a) Model shape, waving arm sequence

(b) Target shape, waving arm sequence

(c) Model shape, punch sequence

(d) Target shape, punch sequence

Figure 4.2: Shape representation for the examples provided in this chapter. Note the heavy clutter resulting from overlapping exposures.

only work for multi-exposure images without any overlap between consecutive exposures. In case of overlapping exposures, color information alone does not provide enough constraints to solve for the desired transformations. Since we wish to explicitly allow for overlapping exposures, we consider only the shape of the object, described by its contour. Contrary to object color/intensity, the contour is not corrupted during the multi-exposure image formation process.

We describe the object depicted in image I_S as a set of n discrete contour points $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, and the collection of exposures in I_M as the set of m points $\hat{\mathcal{P}} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_m\}$, $\mathbf{x}_i, \hat{\mathbf{x}}_j \in \mathbb{R}^2$. Note that the points $\mathbf{x}_i$ are not required to be key points such as those found by a corner detector, nor need they be extrema of a scale space operator such as SIFT features [97]. It is sufficient to sample the points from the output of an edge detector, in our case the result of the Compass operator [123]. Furthermore, each contour point is assigned a local shape context h_i [18], a coarse histogram of relative coordinates of the neighboring points. In the following, we call the tuple $\mathbf{e}_i = (\mathbf{x}_i, h_i)$ of a contour point location $\mathbf{x}_i$ and the associated local shape context h_i an *edglet*. We further refer to the set $\mathcal{E} = \{\mathbf{e}_i | i = 1, \ldots, n\}$ as the *model shape* and the set $\hat{\mathcal{E}} = \{\hat{\mathbf{e}}_j | j = 1, \ldots, m\}$ as the *target shape*, see Fig. 4.2 for the examples used in this chapter. Our model is similar in spirit to Active Shape Models proposed by Cootes and Taylor [41]. However, our approach does not rely on a statistical shape representation to restrict the deformation of the model.

As already pointed out by Thayananthan et al. [147], shape contexts become unreliable in cluttered environments. Multi-exposure images suffer from a lot of clutter, introduced by the multiply exposed object. Furthermore, we expect the model shape to deform over a wide range, leading to completely different shape contexts for corresponding model points. Global shape contexts that take the entire image plane into account are therefore not robust for matching multi-exposure images. Instead, we use a very localized shape context, computed from a dozen neighboring points around each point. This minimizes corruption of the shape contexts in heavily overlapping regions, but it still offers reasonable expressiveness for non-overlapping parts of the shape.

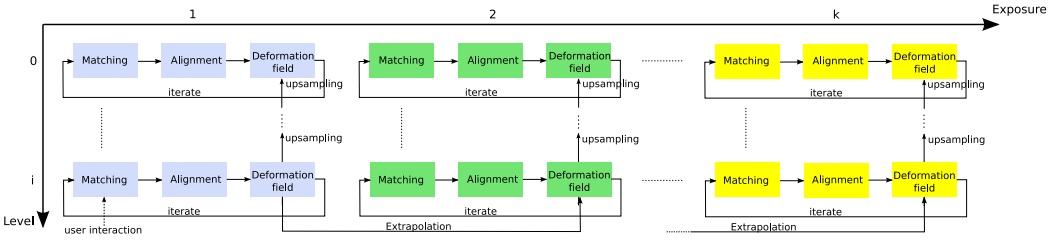


Figure 4.3: Schematic overview of the multi-exposure analysis algorithm. It starts at a low resolution level and iterates until convergence for the first exposure on the highest resolution level is achieved. We then extrapolate the resulting deformation to initialize our algorithm for the second exposure and start over again. This process is repeated until convergence for all k exposures is reached.

4.4 Algorithm

Our multi-exposure analysis algorithm consists of three main parts: correspondence estimation, computation of a set of aligning transformations, and construction of a dense deformation field. These three steps are repeatedly iterated until convergence for a certain exposure is reached. Inspired by coarse-to-fine optical flow proposed by Bergen et al. [19], we employ our algorithm in a multi-resolution framework, i.e., we apply it to an image pyramid starting at low resolution. This way, we are able to quickly find a coarse alignment of the model shape and a given exposure of the target shape, which is a vital condition for distance-based matching at higher resolutions. Results are transferred to the next higher level in the hierarchy by up-sampling the deformation field. The algorithm is repeated until all exposures are matched. A schematic overview is given in Fig. 4.3. The subroutines are explained in the following subsections.

4.4.1 Matching shapes

Given a model and a target shape, we first need to find a mapping Φ from the set of model shape edglets $\mathcal{E}$ to the set of target shape edglets $\hat{\mathcal{E}}$, i.e., $\Phi : \mathcal{E} \rightarrow \hat{\mathcal{E}}$. We require the mapping Φ to be one-to-one. This is the well-known assignment problem which can be encoded in a graph structure [42].

In our setup, the graph consists of $2n + m$ nodes, Fig. 4.4; one node for each edglet in $\mathcal{E}$, one node for each edglet in $\hat{\mathcal{E}}$, and one additional occluder node for each edglet of the model shape. The occluder nodes allow for one-to-one matching without the need to match an actually existing edglet, making it possible to match also edglet sets of

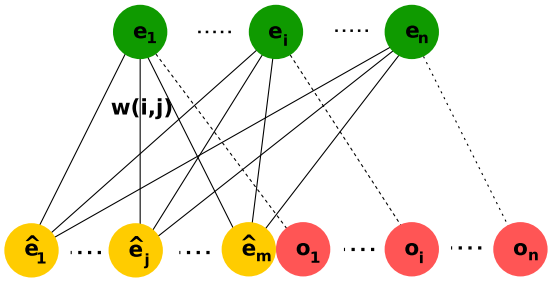


Figure 4.4: The assignment graph. Each node of the model shape (green) is connected to each node of the target shape (yellow). Furthermore, each node of the model shape is connected to its occluder node (red).

unequal sizes. For each node of the model shape, we add a graph edge to each node of the target shape. Furthermore, each model shape node is connected to its occluder node, Fig. 4.4. Finally, each edge between a model and a target node is assigned a weight, modeled by the following formula

$$w(\mathbf{e}_i, \hat{\mathbf{e}}_j) = -(1 + \delta(d_{ij}, \chi^2_{ij})) \cdot D(d_{ij}) \tag{4.1}$$

where

$$\begin{aligned} d_{ij} &= \|\mathbf{A}_l \mathbf{x}_i - \hat{\mathbf{x}}_j\|^2, \\ \chi^2_{ij} &= \frac{1}{2} \sum_k \frac{[h_i(k) - \hat{h}_j(k)]^2}{h_i(k) + \hat{h}_j(k)}, \\ D(x) &= \frac{a}{1 + e^{-bx}}, \\ \delta(x, \chi^2) &= \begin{cases} 0, x \leq c \\ \chi^2, \text{otherwise} \end{cases}. \end{aligned}$$

a and b are chosen such that the maximal cost for the Euclidean distance is limited by a. The value for c is found empirically and is set to 20. $\mathbf{A}_l$ is the locally restricted aligning transformation for translet l, cf. Ch. 4.4.2, and is initialized to the identity matrix. The edges to the occluder nodes are treated separately and are assigned a fixed weight. This weight models the maximally feasible matching distance and can be used to restrict the matching. By combining shape context distance and Euclidean distance into the weight function Eq. (4.1), we combine the advantages of both matching approaches. When the distance between edglets is smaller than the threshold c, the influence of the shape context is discarded, i.e. the matching is driven by Euclidean

distance. Vice versa, if the distance is larger, shape context similarity becomes important and helps to distinguish wrong matches from correct ones. Small distances between matching edglets usually occur in heavily overlapping regions, regions where shape context-based matching is problematic; larger displacements potentially lead to less overlap and hence to an increase in the reliability of localized shape contexts. Furthermore, small displacements also mean that parts of the shape are already coarsely aligned, further justifying a matching based on Euclidean distance.

The solution to the assignment problem we are searching for maximizes the benefit over all assignments $\mathcal{S}$,

$$\max_{s \in \mathcal{S}} \sum_{i=1}^{n} w(\mathbf{e}_i, \Phi_s(\mathbf{e}_i)).$$

An optimal assignment is then found by applying the auction algorithm [21] to the graph constructed above. The auction algorithm is used to solve the assignment problem because its time complexity [21] ($\mathcal{O}(nA\log(nC))$) is better than that of the Hungarian method [113] ($\mathcal{O}(n^3)$) which is typically used to solve this type of problem. Pairwise matching methods such as spectral matching [85] are not applicable in this case given the sheer number of edglets to be matched. While the assignment found by the algorithm is optimal in terms of the weight function Eq. (4.1), the resulting matching still contains outliers. However, we do not aim at producing optimal matches (considered from the point of view of a human observer) within one iteration of the matching algorithm. Instead, we apply several iterations of the algorithm, interwoven with the estimation of a set of aligning transformations described in the following section, Fig. 4.3. Furthermore, the user initially has the possibility to influence the matching by defining preferred edglet matches.

4.4.2 Aligning shapes

Given a mapping Φ, we now want to estimate a set of aligning transformations that map each feature to its found correspondence. Methods proposed in the literature such as thin-plate spline interpolation [25] or moving-least squares based interpolation [126] try to find a globally smooth transformation for the given set of correspondences. However, a globally smooth solution is too restrictive, e.g., such methods offer no way to account for discontinuities and can be prohibitively expensive to compute given more than a thousand feature pairs in our setup [126].

Instead, our goal is to construct piecewise local transformations that only map a certain subset of edglet correspondences onto each other and that allow to preserve discontinuities in the field. To this end, we first have to decide how many transformations are needed to faithfully align the shapes. In order to start with a good guess for the number of transformations, we compute a superpixel segmentation [55] of the single-exposure image I_S. This yields a conservative over-segmentation of the image where each segment in turn is a consistent unit, i.e, all pixels in a superpixel are most similar in color and texture and are thus likely to move under the same transformation. The size of a superpixel is typically in the order of twenty pixels.

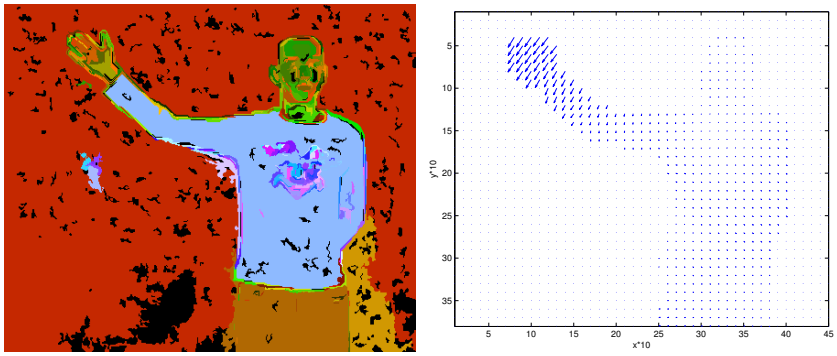
Each superpixel is then assigned the set of edglet matches that lie within its image region. We call this combination of a superpixel and a set of edglet matches a *translet* $\mathbf{t}_l$. Next, we estimate a transformation matrix $\mathbf{A}_l$ for each translet $\mathbf{t}_l$ if it contains a minimal number of edglet matches. In order to robustly estimate the transformation in the presence of potential mismatches, we apply the RANSAC algorithm to find the best transformation and to further classify the set of edglet matches into inlier and outlier matches based on the matching error

$$\epsilon_{i,j} = \|\mathbf{A}_l \mathbf{x}_i - \hat{\mathbf{x}}_j\|.$$

There is no restriction on the kind of transformation, but in our experiments perspective transformations turned out to give the best results. Based on the estimated transformation, we compute an error measure for the current segmentation,

$$E_{Seg} = \sum_l \sum_{(\mathbf{e}_i, \hat{\mathbf{e}}_j) \in \mathbf{t}_l} \epsilon_{i,j}. \tag{4.2}$$

The next step is to optimize the superpixel segmentation to better represent the distribution of local perspective transformations and to minimize the accumulated error E_{Seg} in Eq. (4.2). Since the initial superpixel segmentation usually results in a strong over-segmentation, we are able to combine neighboring translets with similar transformations. In theory, it may sometimes be necessary to split a translet; in our experiments, however, this case rarely occurred, and its influence can be safely neglected. The optimization is carried out in a greedy manner: in each iteration, we find the neighboring translets that yield the highest improvement for Eq. (4.2), collapse the



(a) Translet map (b) Corresponding dense field

Figure 4.5: The translet map (a) encodes the region of influence of each transformation. Black areas do not belong to any translet since they contain no edglets. Based on those transformations, we construct a dense deformation field, shown in (b).

two, and estimate a new transformation for the resulting translet. This greedy optimization is repeated until either the accumulated error reaches a minimum, or a given minimal number of translets is obtained.

At this stage, we have a set of translets, each defining a locally optimal perspective transformation for the given matches. It further defines the regions of influence of each transformation. Fig. 4.5(a) shows the final translet map for the first exposure of the multi-exposure image shown in Fig. 4.1. Regions of the same color encode the region of influence of a translet. Black areas do not belong to any translet since they contain no edglets.

4.4.3 Deriving a dense deformation field

So far, the correspondences only define a sparse deformation field. Since we want to transform the entire shape, we have to construct a dense deformation field from the set of sparse correspondences. The construction should smooth small differences between neighboring translets with similar transformations while preserving large differences. Furthermore, the resulting deformation should interpolate the estimated correspondences. To this end, we initially consider each translet in the translet map, Fig. 4.5(a), and compute per-pixel displacement vectors using the estimated transformation. In a second step, we smooth the resulting dense field by solving an anisotropic nonlinear diffusion PDE [159] on the entire image plane. In order to enforce consistency with

the estimated matches, we treat the corresponding displacement vectors as boundary conditions to the diffusion process. By applying nonlinear diffusion, we smooth out small differences while preserving discontinuities in the field, Fig. 4.5(b).

4.4.4 Avoiding multiple matches in the multi-exposure shape

After the algorithm has converged for a certain exposure, we need to decide what to do with the matched edglets in the target shape. Some of them can be removed since they only belong to the current exposure, others may belong to more than one exposure, e.g., edglets belonging to a region of the scene that doesn't move. These edglets have to be kept in order to provide good alignments for the following exposures. Therefore, instead of removing edglets from the target shape, we penalize matches to already matched edglets in all further iterations of the algorithm. We set the penalty to be a multiple n of the matching distance and extend the weight function given in Eq. (4.1) to

$$\tilde{w}(\mathbf{e}_i, \hat{\mathbf{e}}_j) = w(\mathbf{e}_i, \hat{\mathbf{e}}_j) - n \cdot \|\mathbf{x}_p - \hat{\mathbf{x}}_j\|^2, \tag{4.3}$$

where $\mathbf{x}_p$ is the source of the previous match for target edglet $\hat{\mathbf{x}}_j$. We extrapolate the final deformation field for the current exposure to get a good starting point for the next exposure. This also results in a speed-up since fewer iterations are necessary to converge to a stable matching.

4.5 Results

We present results for fast motion sequences carried out by human actors. All images are taken using a Canon EOS 5D digital camera, equipped with a 28mm prime lens. The environment has to be completely dark, the only source of illumination being a high-output stroboscope. The exposure time of the camera is set such that several flashes are recorded, and the entire motion sequence is mapped in one image. The frequency f of the stroboscope is used to control the number of exposures captured in the multi-exposure image. For the multi-exposure images shown in Figs. 4.6 and 4.7 the frequency of the stroboscope was set to $f = 45Hz$. The single-exposure image should be taken around $\frac{1}{f}$ seconds before the multi-exposure to simplify extrapolation.

We present results for the multi-exposure images shown in the first rows of Figs. 4.6 and 4.7. The images show the recovered deformation fields for each exposure, and their

application to the single-exposure image for three different time instants. Figs. 4.7(c) and 4.8(c) show a synthesized motion-blurred image computed from interpolated intermediate time instants for both examples. As can be seen in Figs. 4.6, 4.7, our method is capable of reconstructing dense motion vector fields from multi-exposure images displaying a smooth and continuous motion. The motion vector fields yield plausible results when applied to the single-exposure image. Fine details as the fingers in the waving arm sequence and the highlights on the balloon of the punch sequence are well preserved.

4.5.1 Limitations

Concerning the accuracy of our method, we are aware that it is not comparable to optical flow methods. Our algorithm yields correct per-pixel displacement vectors only for those pixels identified as edglets. Fine details which are not covered by the shape representation are only coarsely aligned, for example the fingers in Fig. 4.8(l).

Our method works for directed, continuous and smooth types of high-speed motion, i.e., types of motion where the temporal ordering induces a spatial ordering in the image plane. The single-exposure image used to initialize the proposed method needs to obey that ordering constraint, too. It also has to fit into the sampling pattern of the multi-exposure image. The latter requirement is due to the fact that we extrapolate the deformation field to get a good initialization for the multi-exposure image.

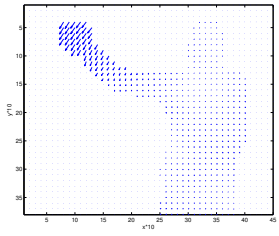
Since our method relies on linear photometric superposition of exposures, it is restricted by the dynamic range and the sensitivity of the camera's sensor. With higher repetition rate, the duration of each flash, and hence the amount of light, decreases. With too many exposures, sensor elements start to saturate. Both effects result in a decrease in contrast in the multi-exposure image and lead to missing edges in the shape representation until the shapes can no longer be aligned. Experiments have shown that for multi-exposure images exhibiting a strong overlap, up to eight exposures can be robustly detected. The problem of faint flashes coming with high repetition rates of the stroboscope can also be countered by increasing the sensitivity of the camera sensor. However, this also increases camera noise which falsifies edge detection, ultimately leading to wrong displacement estimation. Further, the increased sensitivity of the sensor will also lead to a faster saturation of sensor elements, allowing to capture fewer exposures.

Figure 4.6: Resulting deformation fields for the waving arm sequence. The first row shows the multi-exposure, the single-exposure image and a synthesized motion blurred image of the sequence (left to right). The following rows show the recovered deformation fields for each exposure in the first column. The second and third column show the interpolated result created by applying the deformation field to the single exposure to obtain different time instants. Note that the time instants in the second column are not recorded in the original images.

(a) Multi-exposure, waving arm sequence

(b) Single-exposure, ($t = 0$), waving arm sequence

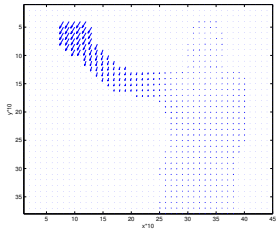
(c) Synthesized motion-blurred image



(d) Deformation field for $t = 1.0$

(e) (c) warped to $t = 0.5$

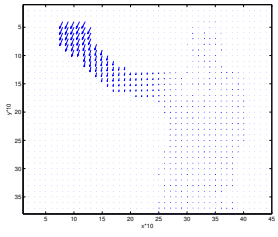
(f) Result at $t = 1.0$



(g) Deformation field for $t = 2.0$

(h) Result at $t = 1.5$

(i) Result at $t = 2.0$

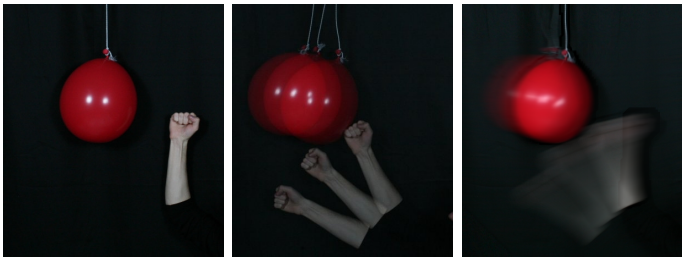


(j) Deformation field for $t = 3.0$

(k) Result at $t = 2.5$

(l) Result at $t = 3.0$

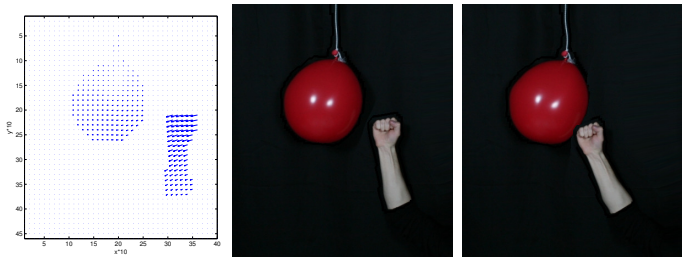
Figure 4.7: Resulting deformation fields for the punch sequence. Images are arranged as in Fig. 4.6. In this sequence, the first matching for the first exposure was guided by specifying a few edglet matches in the contact area of the fist and the balloon.



(a) Single-exposure ($t = 0$), punch sequence

(b) Multi-exposure, punch sequence

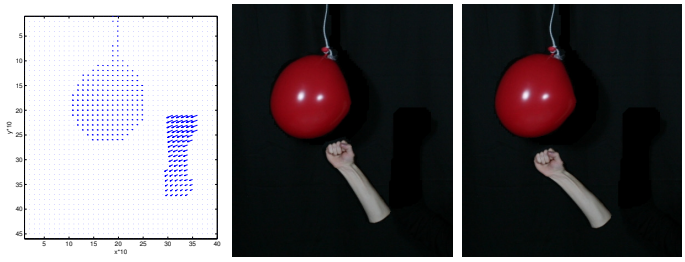
(c) Synthesized motion-blurred image



(d) Deformation field for $t = 1.0$

(e) Result at $t = 0.5$

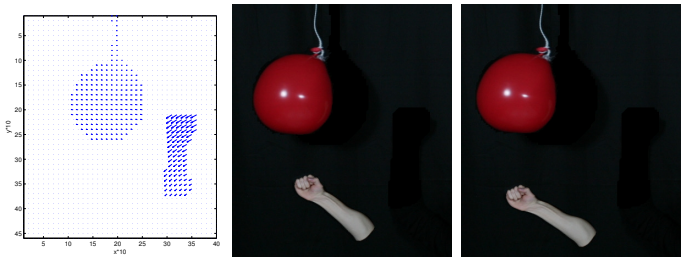
(f) Result at $t = 1.0$



(g) Deformation for $t = 2.0$

(h) Result at $t = 1.5$

(i) Result at $t = 2.0$



(j) Deformation for $t = 3.0$

(k) Result at $t = 2.5$

(l) Result at $t = 3.0$

4.6 Conclusion

In this chapter, we presented an approach to estimate dense motion vector fields from a multi-exposure image of high-speed motion. We demonstrated the applicability of our method to real-world non-rigid scenes. We discussed the capabilities of our method and pointed out its current limitations.

We believe that our work is a step towards image-based analysis of fast motion events without the need for special hardware other than a stroboscope. An interesting avenue for future work could be the integration of our approach into an analysis-by-synthesis loop. Given the warped images, we believe that our results can be refined by synthesizing a multi-exposure image from the warped single exposures and minimizing the residual to the recorded multi-exposure image. Finally, extending our approach to multiple views in order to estimate dense 3D motion vector fields would be an interesting extension for future work.

5

Dense Correspondence Estimation for Image Interpolation

5.1 Introduction

This thesis concerns space-time viewpoint navigation based on recently proposed image interpolation techniques [139, 140]. At the very heart of interpolation techniques lie dense correspondence fields. In Ch. 4, we derived dense correspondence fields from a sparse set of edge correspondences in a multi-exposure setting. Now, we consider conventional video recordings as input.

The edge-based strategy of Ch. 4 can also be applied to two single exposure image as has been shown by Stich et al. [139]. In Refs. [141] and [140], Stich et al. compared their approach to interpolation sequences generated from dense correspondence fields computed with well-known optical flow algorithms [31, 73]. While the proposed edge-based correspondence estimation technique yields perceptually higher quality interpolation sequences, Stich et al. do not compare their results against interpolation sequences generated from motion fields of state-of-the-art optical flow research as reflected by the Middlebury evaluation database [12].

In this chapter, we evaluate the current state-of-the-art in optical flow research for image interpolation sequences. In contrast to the Middlebury evaluation database [12], our focus is not on numerical accuracy of the obtained flow fields, but on perceptual

plausibility of the interpolation sequences. The rest of this chapter is structured as follows: we start with an in-depth review of the state-of-the-art, Ch. 5.2. This is followed by a brief overview on applications of optical flow algorithms in a computer graphics context in Ch. 5.3. After that, we evaluate the current state-of-the-art in optical flow algorithms on several real-world scenes used in the Virtual Video Camera system in Ch. 5.4. We summarize our evaluation and draw conclusions in Ch. 5.5.

5.2 Related Work

Dense correspondence estimation has a long standing history in computer vision research, and a huge number of papers on different aspects of the problem have been published. Since a complete survey on all optical flow algorithms is out of the scope of this thesis, we refer the interested reader to Refs. [3, 14, 16, 107, 111, 117, 142] for previous surveys on the state-of-the-art instead. In the following sections, we will focus on more recent optical flow methods, especially on those ranked on the Middlebury optical flow page [12] since those can safely be considered to represent current state-of-the-art. Nevertheless, we also include the discussion of two older methods, i.e. Horn-Schunck and Lucas-Kanade, for their popularity in the computer graphics community and their availability in several open source libraries. We follow a classification introduced in Ref. [6] and extend it suitably.

5.2.1 Differential methods

Differential methods are based on a first degree approximation of the brightness constancy assumption, i.e.

$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} = 0. \tag{5.1}$$

Horn-Schunck. Eq. (5.1) can be solved globally by adding an additional regularization term to the under-determined system and enforcing global smoothness. This is the classic Horn-Schunck approach [73]. Their idea is to minimize

$$\int_\Omega \left((\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t})^2 + \alpha(|\nabla \delta x|^2 + |\nabla \delta y|^2) \right) dxdy.$$

Ω denotes the image domain and α weighs the influence of the smoothness term.

Lucas-Kanade. Lucas and Kanade [98] take a different approach: they solve Eq. (5.1) for every pixel by assuming all pixels (x, y) within a fixed-size window move with the same flow and construct an over-determined system of equations,

$$w(x,y)\left(\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\right)\Big|_{x,y,t} = 0.$$

Since the system is over-determined, δx and δy are found as a weighted least squares solution. The weights $w(\cdot)$ diminish the influence of pixels farther away.

Combined local-global approach. Often, local methods such as Lucas-Kanade are more robust against noise, while global methods such as Horn-Schunck yield dense flow fields inside homogeneous regions. Bruhn et al. proposed a combined local-global (CLG) approach that incorporates the advantages of both paradigms [32], : It is highly robust under Gaussian noise while giving dense flow fields.

5.2.2 Multi-scale

The linear approximation made in Eq. (5.1) is only valid for small displacements. A common solution to estimate larger displacements is to use a multi-resolution coarse-to-fine approach. An image pyramid is constructed by repeatedly downsampling the image [15, 54]. The optical flow can then be computed on the coarsest resolution, and an upsampled version of δx and δy is used to initialize the solution on the next finer level. This process is then iterated until the final image resolution is reached.

Usually, the image is downsampled by a factor of two. It has long been believed that the performance of multi-resolution algorithms can be further improved by controlling the spatial frequency content more finely than with power of two image pyramids. This is usually achieved by filtering the individual pyramid level with a low-pass filter to generate different scales [31]. However, Sun et al. recently showed that the influence of the downsampling factor on flow quality does not have any statistical significance [143].

Further, multi-scale methods are limited by object size: if the object motion is larger than the extent of the object itself, the object will be smoothed away before its motion can be estimated.

5.2.3 Variational methods

Recent work on optical flow has mostly turned to variational approaches, seeking to minimize an energy functional over the entire image domain Ω. While differential methods linearize the optical flow equation already in the problem formulation, Brox et al. perform a non-linear optimization by postponing linearization to the numerical scheme [31]. Expressed in equation form, the functional reads

$$E(\mathbf{v}) = \int_{\Omega} (\underbrace{\sum_i \beta_i \Psi_i(D_i)}_{\text{data term}} + \alpha \underbrace{\Phi(\mathbf{v}, \nabla\mathbf{v}, \ldots)}_{\text{smoothness term}}) dxdy,$$

where D_i denotes the data term, usually some function of the input images, Ψ_i and Θ are robust functions to penalize outliers.

Data terms. The data term typically takes the form

$$\Psi_i(D_i) = \Psi_i(\mathcal{L}_i(I(x+\delta x, y+\delta y, t+\delta t)) - \mathcal{L}_i(I(x,y,t))).$$

The linear function $\mathcal{L}_i$ (e.g., identity, gradient, Laplacian) allows to generalize the gradient constancy assumption to include other constancy assumptions as well. Substituting $\Psi(x) = x^2$ and assuming the identity function for $\mathcal{L}_i$, one arrives at the classical Horn-Schunck data term. This data term has been extended by Brox et al. to also include the image gradient in the data term [31]. Going beyond intensity and gradient differences, Wedel et al. also integrated the fundamental matrix F into the data term [158]. If the scene is static, this helps to further restrict possible flow vectors in texture-less regions. While the classical approaches use the L^2-norm as penalty function in the data term, Brox et al. propose to use the robust Charbonnier penalty $\Psi(s^2) = \sqrt{(s^2 + \epsilon^2)}$ to robustify flow computation [31]. Recently, researches also tend to use $\Psi(x) = |x|$ in the data term [162, 175]. This robustifies the flow computation against illumination changes, occlusion and noise. Unfortunately, this norm is not differentiable and thus difficult to use.

Methods of regularization. Much research has been devoted to finding a good regularization strategy to fill in missing regions. The classical approach of Horn-Schunck is to use the L^2-norm of the flow field gradient as regularizer. It yields a convex functional that can be optimized globally and strongly penalizes discontinuities in the optical flow

field. This usually leads to blurry flow fields around edges. Starting with the approach of Black and Anandan [24], different robust functions have been proposed for regularization. However, some of them are non-convex and thus difficult to optimize.

Based on this approach, Zach et al. propose to use a true total variation (TV) regularizer [175]. This helps to better preserve discontinuities in the flow fields. Along with the use of the TV-regularizer, they extend an efficient projected gradient scheme proposed by Chambolle [37] that allows optical flow computation in real-time. Trobin et al. also considered the problem of piecewise constant flows in untextured regions [150]. Instead of applying a TV-L^1 regularization to the flow field, they propose to use an unbiased second-order regularizer to remove the bias towards constant flow fields. Werlberger et al. extended the isotropic Total Variation regularization approaches to an anisotropic regularization based on the robust Huber-norm [162]. Similar, Wedel et al. propose to use data-aware regularizers that adaptively favor rigid body motion if supported by the image data and motion field discontinuities that coincide with discontinuities of the image structure [157].

While all of the approaches so far focused on optimizing the data fidelity term and the smoothness term separately, Zimmer et al. recently investigated the interplay of these two terms [179]. Inspired by an early model by Nagel and Enkelmann [115], which regularizes the flow field along edges but not across, they develop a synergistic model where data term and smoothness term do not interfere but complement each other instead. Their anisotropic smoothness term reduces smoothing in the data-constraint direction, while enforcing a strong filling-in effect orthogonal to it.

A different approach to tackle occluded and texture-less regions is taken by Xu et al. [172]. In their work, they infer a segment-based affine motion model using a color segmentation and an initial flow field obtained by Brox's method. After having inferred an occlusion map and a confidence map from the segments, the parametric motion model is incorporated in a variational framework, and the initial flow field is refined. Zitnick et al. also jointly estimate optical flow and segmentation and further allow fractional contributions of overlapping segments to individual pixels [180]. Based on the Gestalt principles of grouping, Werlberger et al. recently also incorporated a low level image segmentation into flow estimation [161].

Efficient minimization strategies. The minimization of the variational approaches is usually carried out in an iterative fashion, applying for example a projected gradient scheme. A drawback of such local iterative minimization techniques is their slow convergence and the risk of getting stuck in local minima. To account for this, Trobin et al. propose to approximate the minimization by solving a series of binary subproblems to facilitate large optimization moves [149]. Their proposed method can be interpreted as an extension of discrete graph-cut based methods such as α-expansion [28] or LogCut [84] to a spatially continuous setting.

5.2.4 Long-range methods

Despite the wide-spread use of multi-resolution methods in optical flow estimation, there are still cases where the displacement is too large to be estimated in a hierarchical framework. This is especially true if the object motion is larger than the object itself. Multi-resolution methods will not help in this case, since the object will vanish in the image pyramid before the displacement is small enough to be estimated. To account for this, Brox et al. recently incorporated descriptor matching in a variational framework to guide optical flow estimation for larger motions [30]. As descriptors, they propose to use regions descriptors of a hierarchical segmentation of the image, similar to the SIFT descriptor [97]. Their approach combines the power of descriptor matching with the regularization properties of a variational approach.

A different approach has been taken by Steinbrücker et al. [136]. Starting at a standard variational formulation and making use of techniques known from quadratic relaxation, they arrive at a formulation with a point-wise data term and a convex smoothness term which are coupled via an additional flow field. For both data and smoothness term, a globally optimal solution can be found. The solution for the data term can simply be computed by a complete search, alleviating the need for coarse-to-fine warping strategies. Another appealing property of this approach is that any point-wise error measure can be integrated into the data term. This has been exploited in [137] where they integrate patch-based error measures into this framework.

While both approaches clearly improve on the current state-of-the-art for long range motions, they both suffer from a lower overall accuracy compared to warping-based methods [30, 136].

5.2.5 Discrete optimization

While variational methods seek to minimize the energy functional in a continuous domain, there are also approaches to optical flow computation using methods from discrete optimization. A common approach to optical flow estimation in a discrete setting is to formulate the process as maximum a-posteriori (MAP) inference. Contrary to continuous approaches, the solution is no longer continuous-valued but requires a sampling of the solution space. To this end, discrete methods usually have to deal accuracy for a computationally tractable label space.

Glocker et al. overlay the image with a uniform grid of control points and iteratively estimate displacement vectors for each control point based on Markov Random Fields and a warping strategy [58]. To account for the limited precision of a discrete label space, they estimate the uncertainty of the flow field in each control point and derive a new label space for every control point in every iteration. The dense optical flow field is obtained via cubic B-spline interpolation of the control points.

Lempitsky et al. combine discrete and continuous flow estimation [83]. Their algorithm fuses multiple proposal optical flow fields obtained using continuous optical flow estimation algorithms such as Horn-Schunck and Lucas-Kanade. Using a graph-cut optimization to decide for each pixel which proposal flow field the flow vector should be taken from, they achieve lower energy values than with either a pure discrete or purely continuous optimization approach.

5.2.6 Learning-based methods

Despite the long history of optical flow computation and the study of various data and smoothness terms, very few attempts have been taken to optimize those terms from actual data. In one of the first approaches to supervised learning optical flow [122], a detailed analysis of flow statistics in natural scenes is presented, and machine learning methods are developed to learn a Markov random field model of optical flow. The prior probability of a flow field is formulated as a Field-of-Experts model that captures the spatial statistics in overlapping patches and is trained using contrastive divergence. This model is extended in Ref. [144] to the spatio-temporal domain to model temporal changes in image features. Further on, in this approach the statistical relationship

between image and flow boundaries are modeled explicitly by a Steerable Random Field following the model proposed by Nagel and Enkelmann [115].

Instead of considering maximum likelihood estimation, Li and Huttenlocher [87] learn the parameters of a continuous-state Markov random field by minimizing the training loss for a set of ground-truth images. They use a technique from stochastic optimization, called simultaneous perturbation stochastic approximation, to optimize the error criterion used to evaluate the quality of the flow field. Their approach does not require approximations common in maximum-likelihood estimation and generalizes well to unseen data.

5.2.7 Occlusion handling

Occlusion handling is an important aspect of optical flow computation since no sensible correspondences can be found for occluded regions. If disregarded, the flow fields along occluding boundaries tend to collapse due to regularization. A first step towards accounting for occlusion was taken by jointly estimating forward and backward optical flow fields [5]. By jointly estimating both flow directions, occluded regions can be identified by examining the mismatch of forward and backward flow. Another approach to occlusion detection is to compute the divergence of the flow field and looking for areas with negative divergence. Sand and Teller [125] combine this approach with pixel projection differences to detect occluded regions and integrate this into the variational method by Brox et al. [31]. In their approach, they alternate optical flow estimation and occlusion detection. Similarly, Xiao et al. alternate optical flow computation and occlusion detection based on intensity mismatch [170]. Occluded regions are filled by adaptive bilateral filtering of the flow fields. In contrast to this, Ince and Konrad simultaneously estimate optical flow and occluded regions in a variational framework [76]. Optical flow in occluded regions is inpainted from neighboring visible regions using image-driven anisotropic diffusion.

A different approach is taken by Sellent et al. [132]: they propose an image formation model that relates a long-exposure image to preceding and succeeding short-exposure images in terms of optical flow and occlusion. With this method, not only binary occlusion maps but also the per-pixel occlusion time can be recovered. While originally also a two-step process, it has been recently integrated in a variational framework that allows simultaneous estimation of flow and occlusion information [131].

5.2.8 Performance evaluation

With a huge number of different optical flow algorithms available, datasets for qualitative evaluation of those algorithms have become an essential part of research. Starting with the benchmark set introduced by Barron et al. [14], the accuracy of optical flow algorithms has rapidly increased. However, after 13 years of research, the performance improvements on those benchmarks have largely saturated. To this end, Baker et al. recently established a new publicly available database [12] which focuses on current aspects of optical flow research, i.e. photo-realistic scenes with all artifacts of real sensors (noise, motion blur, etc.). This database has become a standard by now and it ranks the currently best optical flow algorithms with respect to different error measures. Nevertheless, slight criticism on the design of the test scenes recently arose in the computer vision community. Vaudrey et al. have shown in their report that high ranking algorithms often fail for real-world scenes, e.g. driving scenes with changing illumination and large motions [151].

In the following sections, we evaluate current optical flow research in the specific context of image interpolation, focusing on typical real-world scenes.

5.3 Optical Flow in Computer Graphics - A Survey

In recent years, image-based rendering techniques have advanced from static to time-varying scenes. Instead of still images, multiple, not necessarily synchronized video streams now capture a dynamic scene from different viewpoints. This has lead to several new algorithms which try to extend findings for single images or pairs of images to video sequences. Despite the high processing power in modern PCs, however, it is often unavoidable to propagate some information along the sequence instead of recomputing it every frame in order to keep algorithms efficient. In addition, temporal coherence is an important aspect for visual fidelity of video streams. This section gives an overview of recent applications of optical flow in a computer graphics context.

Image and Video Registration. Image and video registration is the most prominent application of optical flow techniques in contemporary computer graphics. Applications range from image stabilization for hand-held acquisition [77, 146] and registration of video frames with different exposures [50, 51, 160] to registration of projected

textures on the surface of an approximate 3D model [52]. Optical flow has also been used to register different video streams [124] as a preprocessing step for video editing. Most of these approaches rely on a gradient-based variant of the Lucas-Kanade algorithm [98]. As exceptions, Eisemann et al. [52] use the approach by Brox et al. [31], and Einarsson et al. [50] resort to the approach of Black and Anandan [24].

Information Propagation. Another important field of application for optical flow is information propagation along or between video streams. The most prominent application in this area is video matting, where information propagation is used to minimize tedious user interaction [10, 39, 53]. Depending on the required accuracy of the flow fields, authors resort to simple block-matching [53], local flow averaging [10] or the approach of Black and Anandan [39]. Einarsson et al. use flow fields to propagate captured reflectance fields along and between cameras for re-lighting purposes [50]. Peers et al. follows a similar approach to transfer reflectance fields for facial re-lighting [118]. Since both approaches require accurate correspondence fields, they use the algorithms by Black and Anandan [24] and Brox et al. [31], respectively.

Reconstruction and Augmentation. Dense correspondence fields are also needed by reconstruction or augmentation algorithms. Atcheson et al. use the Lucas-Kanade optical flow to extract the distortion in a high-frequency pattern introduced by a heated gas volume [7]. The 2D motion vectors then serve as a basis for the reconstruction of a refractive index field within a volume in a multi-camera setup. Scholz and Magnor [128] also use Lucas-Kanade flow to measure textile motion in a multi-camera setup and to reconstruct the 3D scene flow, which then serves as a basis for the animation of a virtual cloth. Hilsmann and Eisert use optical flow computed on a coarse mesh overlaid on the images to track textile motion in monocular sequences and to augment parts of the textiles with different textures [71, 72].

Image Interpolation. Concerning image interpolation, there are surprisingly few occurrences of traditional optical flow algorithms. Wang et al. use optical flow to warp images of asynchronously captured light fields to a common virtual time before reconstructing the virtual view [153]. Recently, Mahajan et al. proposed a path framework for image interpolation [99]. While the path framework does not compute optical flow

fields in a traditional sense, they show that the paths can be transferred into the traditional optical flow representation. Stich et al. proposed an algorithm for deriving dense correspondence fields from sparse edge matches [140, 141] which serve as input for a perceptual image interpolation algorithm.

In the following section, we evaluate the current state-of-the-art in optical flow research for the task of image interpolation.

5.4 Optical Flow for Image Interpolation - A Case Study

In our case study, we systematically evaluate four different optical flow algorithms. The focus of this study is on the adequacy of the resulting flow fields for use with the Virtual Video Camera, Ch. 3. While optical flow research usually focuses on estimating correspondences from one frame of a camera to the next - in what follows referred to as *temporal image pair* - we are also interested in correspondence fields between frames captured by different cameras, referred to as *spatial image pairs*, and between frames captured by different cameras as well as different points in time, referred to as *spatio-temporal image pairs*. We thus evaluate each algorithm on three image pairs of our test dataset: a temporal pair, a spatial pair and a spatio-temporal pair. Each algorithm has to face several challenges common to correspondence estimation on space-time navigation footage:

1. large pixel distances in the order of up to 20% of the image diagonal, especially for spatial and spatio-temporal image pairs
2. fast moving objects / small objects
3. changing illumination for spatial and spatio-temporal image pairs
4. occlusions and disocclusion due to viewpoint change and motion
5. large untextured regions

5.4.1 Selection of optical flow algorithms

To evaluate the state-of-the-art in optical flow computation, we deliberately chose the two published top performers on Middlebury's flow evaluation data base [12] at the time of this writing.

The algorithm by Sun et al. [143] performs best with respect to the endpoint error, which is considered to be the gold standard for evaluating optical flow accuracy. Sun et al. thoroughly investigated how the objective function, the optimization method, and modern implementation practices influence flow accuracy. They combine their findings with a weighted non-local median filtering term in the classical Horn-Schunck model. The weighted non-local median filtering approach avoids over-smoothing fine image details. With respect to the normalized interpolation error, this algorithms is ranked 19th out of forty.

The algorithm by Werlberger et al. [162] performs best with respect to the normalized interpolation error.This error measure most closely reflects the goal of this study, i.e. interpolation quality. In their approach, Werlberger et al. increase the robustness as well as the accuracy of discontinuity preserving variational optical flow models by replacing the isotropic total variation regularization with an image-driven anisotropic one based on the robust Huber-L^1-norm. They further propose to exploit symmetry around a central frame if more than two images are available.

In addition to the two top performers described above, we also include the algorithm proposed by Steinbrücker et al. [136] in our study. This approach is especially suited for estimating fast, long-range motion without resorting to a coarse-to-fine warping strategy. While pyramidal approaches are able to handle large motions in principle, they still fail as soon as the displacement of the object is larger than the extent of the object itself. This is a well-known limitation of optical flow estimation and has recently been tackled by several researchers independently [30, 136]. We favor the approach by Steinbrücker et al. for its flexibility in the choice of the point-wise data term. This allows to integrate arbitrary point-wise descriptors, e.g. SIFT descriptors [97], into the flow estimation. However, we stick with the original formulation [136] for our comparison.

Finally, we evaluate our test scenes on correspondence fields computed by the algorithm proposed by Stich et al. [140]. The main focus of Stich et al. lies on correctly matching edges and moving regions coherently, inspired by findings from perceptual research. To this end, Stich et al. propose to match edge pixels, followed by a least-squares estimation of a perspective transformation for each image region based on the matches and an initial super-segmentation of the image. Their approach can be consid-

ered a piecewise constant optical flow. While this algorithm also offers the possibility for manual correction of flow fields, we used uncorrected flow fields for fair comparison.

Except for the algorithm by Steinbrücker et al., we used publicly available implementations of the authors. The algorithms by Stich et al., Steinbrücker et al. and Sun et al. operate on color images, whereas for the approach of Werlberger et al., we had to desaturate the images first. The parameters for the different algorithms were set to default values (if provided in the original references) or optimized for the spatio-temporal pair and kept fix for the spatial and temporal pairs. While one can usually achieve better results by tuning the parameters for each image pair individually, we opt for this approach since we want to be able to process thousands of image pairs, rendering individual parameter tuning impossible.

5.4.2 Interpolation method

We evaluate the performance of the selected algorithms by forward-warping both images and adaptively blending them to obtain the interpolated image [140]. The blending weights are determined per pixel from the connectedness of the motion fields [101]. Lacking depth information, Stich et al. propose to use simple heuristics to infer a relative depth ordering from the motion fields. For convenience, we summarize the algorithm in Alg. 1. While there are other interpolation algorithms that are based on backward warping, e.g. the one proposed by Baker et al. [11] and used in the Middlebury evaluation, we stick with the forward-warping approach because it allows also for interpolation from multiple images needed later.

5.4.3 Test scenes

Our test set includes one synthetic test sequence with known ground truth motion fields and depth, and three real-world scenes of varying complexity and different challenges. For the real world data set, ground truth motion as well as ground truth interpolation results are not available but made by hand. The synthetic *Stonemill* sequence features long-range motion of up to 60 pixels at a resolution of 480×270 pixels and large occlusion/disocclusion areas, Fig. 5.1. The *Mona* sequence is a real-world sequence that was captured under controlled studio illumination, Fig. 5.2. It features large displacements of thin structures (arms, hat stands) as well as large untextured regions. The maximal pixel displacement is around 90 pixels at a resolution of 960×540 pixels.

Input: frames I_0, I_1, flow fields u_0, u_1, t
Output: interpolated frame I_t
begin
`// forward warp` I_0
$I'_t(\text{round}(x + t \cdot u_0(x))) = I_0(x)$;
`// forward warp` I_1
$I'_{1-t}(\text{round}(x + (1-t) \cdot u_1(x))) = I_1(x)$;
compute connectedness c_{u_0}, c_{u_1};
`// adaptively blend based on connectedness` c_{u_0} `and` c_{u_1}
$I_t(x) = t \cdot c_{u_0}(x) \cdot I'_t(x) + (1-t) \cdot c_{u_1}(x) \cdot I'_{1-t}(x)$;
end

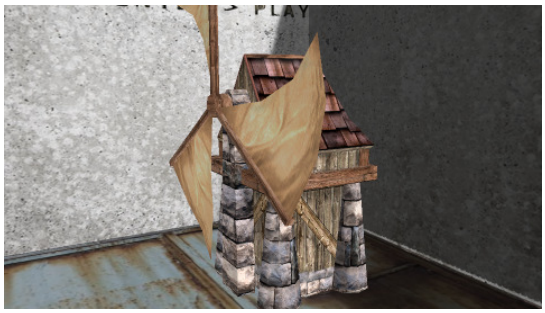
Algorithm 1: Perceptual interpolation algorithm proposed by Stich et al. [140]

The *Skateboarder* sequence adds varying illumination conditions and shadows due to outdoor capture to the set of challenges, Fig. 5.3. The maximal pixel displacement for the tested image pairs is around 180 pixels at a resolution of 960×540 pixels. It further features large disoccluded regions at the boundary and behind the skateboarder. The *Parkour* sequence is the most complex scene in our test set, Fig. 5.4. The complexity arises from the background with a lot of occlusion/disocclusion, the fine structures of the twigs and varying illumination. The challenges along with the image resolution and maximal displacements are listed in Table 5.1.

5.4.4 Evaluation

We compute correspondence fields for all image pairs of our test set using the four selected optical flow algorithms. In addition, we generate ground truth motion fields for the real-world scenes by hand. Ground truth in this case reads as "flow fields producing the visually most plausible interpolation sequence", not necessarily the physically correct motion vectors. We then interpolate an image sequence between the two source images using Alg. 1 and evaluate the quality. All scenes are evaluated in a perceptual user study including interpolation sequences generated from ground truth flow fields; the synthetic sequence is evaluated numerically as well using synthesized ground truth images. A numerical evaluation of the interpolation error on the real-world sequences would only be possible for the temporal image pairs since each camera provides enough frames along the temporal dimension for a leave-one-out comparison. The spatial and


(m) Temporal pair

(n) Spatial pair

(o) Spatio-temporal pair

Figure 5.1: Image pairs used for evaluation on the Stonemill scene.

(a) Temporal pair

(b) Spatial pair

(c) Spatio-temporal pair

Figure 5.2: Image pairs used for evaluation on the Mona scene.


(a) Temporal pair


(b) Spatial pair


(c) Spatio-temporal pair

Figure 5.3: Image pairs used for evaluation on the Skateboarder scene.

(a) Temporal pair

(b) Spatial pair

(c) Spatio-temporal pair

Figure 5.4: Image pairs used for evaluation on the Parkour scene.

Scene	Resolution (pixel)	Max. Disp. (pixel)	Challenges
Stonemill	480 × 270	60	long-range motion, large occluded/disoccluded areas
Mona	960 × 540	90	untextured regions, fast motion of fine structures
Skateboarder	960 × 540	180	long-range motion, large occluded/disoccluded areas, varying illumination
Parkour	960 × 540	170	complex background, multiple motion layers, occlusion/disocclusion, varying illumination

Table 5.1: Test scenes used for evaluation along with their specific challenges.

spatio-temporal pairs, however, are sampled too sparsely in the existing test footage; skipping a camera and/or frame for a leave-one-out comparison would result in too large pixel displacements and insufficient image overlap.

Synthetic sequence with ground truth

We first numerically evaluate the quality of the individual algorithms using the synthetic *Stonemill* sequence along with the ground truth correspondence fields. To this end, we compute the error measures *average endpoint error (AEE)*, *average angular error (AAE)*, *interpolation error (IE)* and *normalized interpolation error (NIE)* as proposed in Ref. [11]. The results are summarized in Table 5.2. Figs. 5.5, 5.6, 5.7 show the interpolation result along with a contrast-stretched visualization of the interpolation error. The numerical evaluation confirms the Middlebury ranking: the approach by Sun et al. [143] performs best, on average, with respect to the angular and endpoint error measures. The algorithm by Werlberger et al. [162] ranks highest with respect to the interpolation and normalized interpolation error. The long-range method by Steinbrücker et al. [136] cannot compete with the two top-performers with respect to the numerical accuracy of the flow fields due to the discrete sampling of the motion vectors. The approach by Stich et al. [140] was designed with perceptually plausible

	Pair	Error	Stich[140]	Werlberger[162]	Sun[143]	Steinbrücker[136]
spatial	$\mathbf{w}_{1,2}$	AAE	57.64	7.56	3.86	11.02
		AEE	16.80	2.99	1.19	7.08
	$\mathbf{w}_{2,1}$	AAE	37.45	34.28	3.64	17.57
		AEE	11.44	8.63	1.64	8.80
		IE	40.45	28.57	29.29	34.08
		NIE	4.08	2.93	3.01	3.57
spatio-temporal	$\mathbf{w}_{1,2}$	AAE	42.37	19.64	22.19	11.75
		AEE	12.23	5.38	5.68	5.60
	$\mathbf{w}_{2,1}$	AAE	43.91	15.12	16.55	17.84
		AEE	14.07	4.97	4.86	7.44
		IE	33.45	28.10	29.60	33.30
		NIE	3.05	2.51	2.69	3.12
temporal	$\mathbf{w}_{1,2}$	AAE	12.45	2.31	0.78	4.00
		AEE	1.45	0.18	0.07	0.68
	$\mathbf{w}_{2,1}$	AAE	8.34	2.53	0.82	4.15
		AEE	0.74	0.21	0.06	0.68
		IE	11.26	11.07	11.47	8.19
		NIE	0.91	0.87	0.84	0.54

Table 5.2: Numerical evaluation of synthetic test sequence. Best values are marked in green.

interpolation results as goal. As such, it is expected to perform poorest with respect to the numerical evaluation.

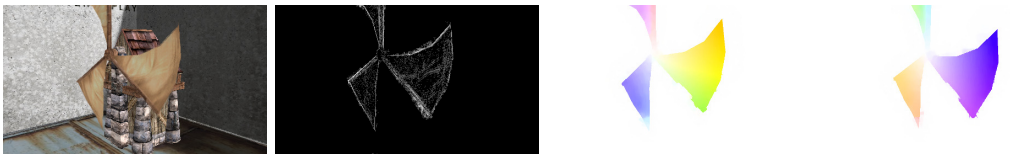
The numerical analysis is restricted to a single frame in the middle of the interpolation sequence. Thus, it naturally masks artifacts arising due to motion along the sequence. However, temporal consistency is an important aspect in terms of interpolation quality. To capture this aspect, we evaluate the visual quality of the whole sequence with a psychophysical user study.

User study

In order to assess the perceptual quality of the interpolation sequences for real-world scenes as well as the synthetic sequence, we followed the approach presented in Ref. [140] and carried out a user study. The two major goals of the study were

Figure 5.5: Interpolation results on the temporal stonemill pair. Rows 2–5 show (left to right) the interpolation result, a gamma-corrected visualization of the interpolation error, and forward and backward flow fields. The approach by Steinbrücker et al. [136] yields the lowest interpolation error, cf. Table 5.2.

(a) Interpolation result using ground truth flow fields. From left to right: ground truth $I_{1.5}$, ground truth flow fields $\mathbf{w}_{1,2}$ and $\mathbf{w}_{2,1}$.



(b) Interpolation result using flow fields generated with Sun et al. [143]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.

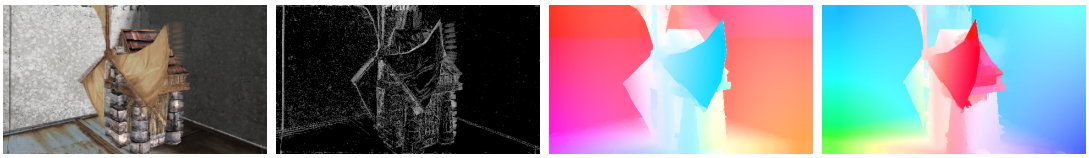
(c) Interpolation result using flow fields generated with Steinbrücker et al. [136]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.

(d) Interpolation result using flow fields generated with Werlberger et al. [162]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.

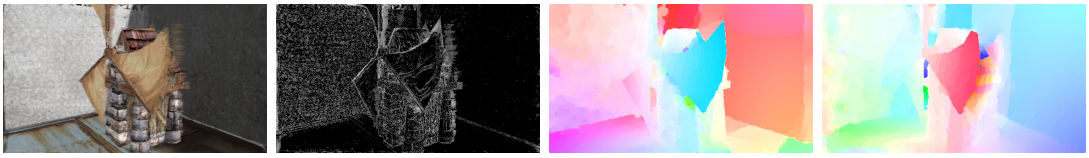
(e) Interpolation result using flow fields generated with Stich et al. [140]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.

Figure 5.6: Interpolation results on the spatial stonemill pair. Rows 2–5 show (left to right) the interpolation result, a gamma-corrected visualization of the interpolation error, and forward and backward flow fields. The approach by Werlberger et al. [162] yields the lowest interpolation error, cf. Table 5.2.

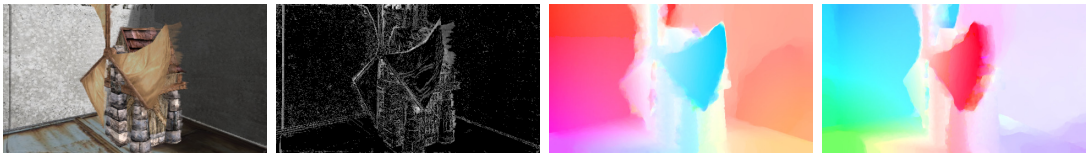
(a) Interpolation result using ground truth flow fields. From left to right: ground truth $I_{1.5}$, ground truth flow fields $\mathbf{w}_{1,2}$ and $\mathbf{w}_{2,1}$.



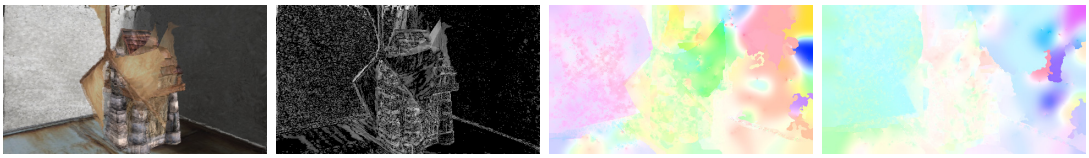
(b) Interpolation result using flow fields generated with Sun et al. [143]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.



(c) Interpolation result using flow fields generated with Steinbrücker et al. [136]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.



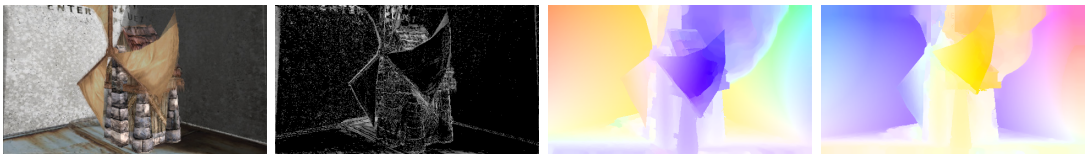
(d) Interpolation result using flow fields generated with Werlberger et al. [162]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.



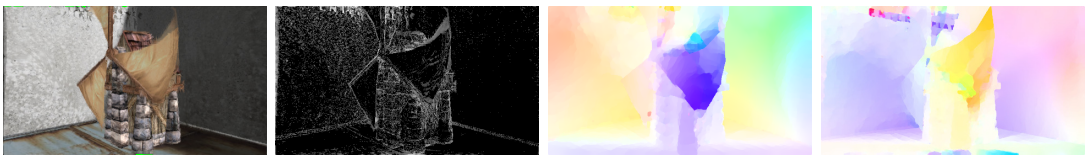
(e) Interpolation result using flow fields generated with Stich et al. [140]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.

Figure 5.7: Interpolation results on the spatio-temporal stonemill pair. Rows 2–5 show (left to right) the interpolation result, a gamma-corrected visualization of the interpolation error, and forward and backward flow fields. The approach by Werlberger et al. [162] yields the lowest interpolation error, cf. Table 5.2.

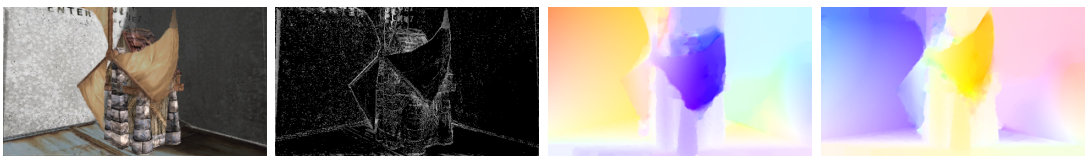
(a) Interpolation result using ground truth flow fields. From left to right: ground truth $I_{1.5}$, ground truth flow fields $\mathbf{w}_{1,2}$ and $\mathbf{w}_{2,1}$.



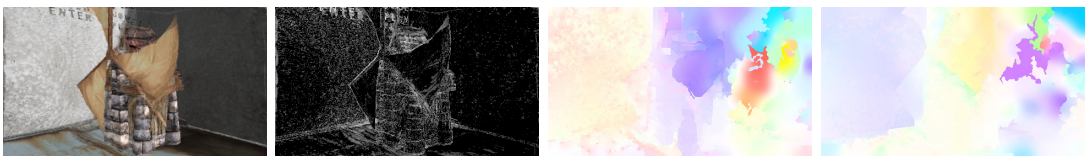
(b) Interpolation result using flow fields generated with Sun et al. [143]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.



(c) Interpolation result using flow fields generated with Steinbrücker et al. [136]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.



(d) Interpolation result using flow fields generated with Werlberger et al. [162]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.



(e) Interpolation result using flow fields generated with Stich et al. [140]. From left to right: $I_{1.5}$, interpolation error, $\mathbf{w}_{1,2}$, $\mathbf{w}_{2,1}$.

(1) to investigate whether any of the tested algorithms attains a perceptual quality score similar to ground truth, and

(2) to compare the results of the tested algorithms against each other and assess whether there is a statistical significance with respect to perceived quality.

Rather than using a standard rating task in which participants would be shown a sequence and be asked to rate its quality, we opted for a more systematic approach. In the psychophysical study, we used a two-alternative-forced-choice task in which two interpolation sequences were shown successively, and participants were asked to indicate which sequence contained less visual artifacts. Such a direct comparison allows for a more fine-grained analysis of the data as rating tasks are often subject to scaling problems. For each of the 4 different test scenes, we compared all 4 different optical flow algorithms as well as interpolations created from ground-truth flow fields against each other (only doing pairwise AB and AA, not BA comparisons), yielding a total of $4 \cdot 3 \cdot (4 \cdot \frac{5}{2} + 5) = 180$ trials. To meet the first goal of our study, we included hand-made correspondence fields as ground-truth into the user study.

All real-world scenes were rendered at a resolution of 960x540 pixels with 25 frames per second and were 2–4 seconds long. Figs. 5.11, 5.12, 5.13 show the interpolation result halfway between the image pairs. The synthetic scene was rendered at a resolution of 480x270 pixels. The sequences were presented on a black-background LCD monitor using a resolution of 1366x768 pixel at 60 Hz. Participants viewed the stimuli at a distance of roughly 50 cm. Each trial consisted of a fixation cross shown for 1 second, followed by the first sequence, a second fixation cross shown for 0.5 seconds, and the second sequence. After this, the screen was blanked and participants were asked to indicate by key press which sequence contained less visual artifacts. Participants were briefed before the experiment that artifacts were defined as "any visual disturbance resulting in non-smooth transition". All participants completed two test trials before the experiment to familiarize them with the task. Neither during the test trials nor the experiment was any feedback given. The whole experiment lasted approximately 35 minutes. Our test group consisted of 9 participants who had *strong* computer graphics-related experience.

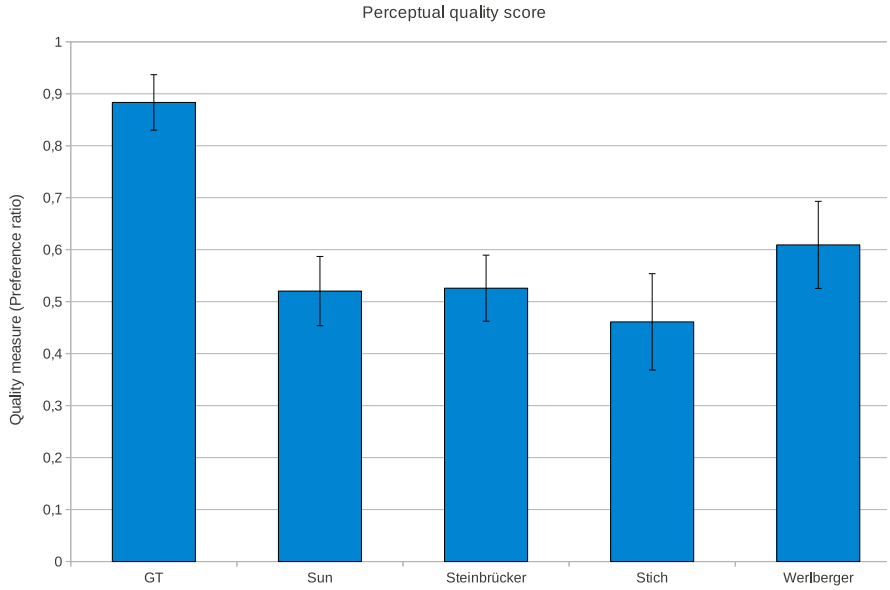


Figure 5.8: Perceptual quality scores for 5 different test conditions (optical flow algorithms).

Analysis

For the first analysis, we determined a perceptual quality score for each algorithm by counting how many times it was chosen as producing fewer visual artifacts when compared to one of the other algorithms. The normalized scores are shown in Fig. 5.8 for all five approaches. As can be seen immediately, there is a significant difference between perceived quality of interpolations based on ground-truth flow fields and perceived quality of all interpolations generated from the other algorithms. This is confirmed by the highly significant one-way anova ($F(4, 40) = 46.546, p = 0.001$). A Tukey test [27] with confidence level $p = 0.01$ further reveals that only the ground truth interpolations differ significantly.

In order to answer the first question of our study, we re-plot the data in Fig. 5.8 to show how often participants would choose any other algorithm over ground truth interpolation, that is, how many times the perceptual quality of the sequence was at least as good as in the ground truth case. We break this analysis down by test scenes. First of all, Fig. 5.9 confirms the results outlined above: none of the four optical flow methods are selected more than 1 or 2 times, on average (out of a maximum of 9),

indicated by the colored dashed lines in Fig. 5.9, over the ground truth interpolation for all scenes. Values around 4.5 in Fig. 5.9 would indicate equal perceived quality. Clearly, none of the tested algorithms can reach this threshold consistently over all tested scenes.

However, with respect to the individual scenes, there are noticeable differences. With respect to the *Mona* scene, the algorithm by Stich et al. [140] provides a perceptual quality comparable to ground truth; the algorithm by Werlberger et al. [162] fares only slightly worse. This scene is extremely well suited for the approach of Stich et al. since most errors are hidden in the large untextured regions in the background, and the important edge structures are distinct and can be matched unambiguously. Also, the algorithm by Werlberger et al. can score on this scene due to its powerful anisotropic regularization. The approach by Sun et al. [143] fails to recover the motion of the arm and also diffuses wrong motion information into background, leading to noticeable artifacts. While the interpolation created from Steinbrücker et al.'s flow fields is able to maintain all fine structures in the background, the visual quality is heavily impaired by wrong matches in the untextured regions between the legs and in the disoccluded region behind the actor's head. Furthermore, the regularization here also diffuses wrong motion information into the background.

Considering the more complex *Skateboarder* scene, no algorithm is able to achieve ground truth quality. For the edge-based algorithm by Stich et al., this scene already exhibits a too complex scene structure where the edge matching approach clearly fails. The algorithms by Sun et al. and Werlberger et al. provide a good interpolation for the foreground object. However, the background gets distorted, and the puddles on the ground move in an unnatural way. Steinbrücker et al.'s algorithm is the only one that manages to transform visible parts of the background correctly, but again suffers from spurious wrong matches that ruin overall interpolation quality.

Surprisingly, for the most complex *Parkour* scene, every algorithm can collect some votes. This might be due to the fact that even the ground truth interpolation suffers from artifacts arising from the depth heuristics employed in the interpolation algorithm. However, only the algorithm by Steinbrücker et al. [136] reaches a perceptual quality comparable to ground truth for the temporal interpolation.

For the synthetic test scene, the approach by Sun et al. [143] is on par with ground truth for the spatial image pair, whereas the algorithm by Steinbrücker et al. outperforms

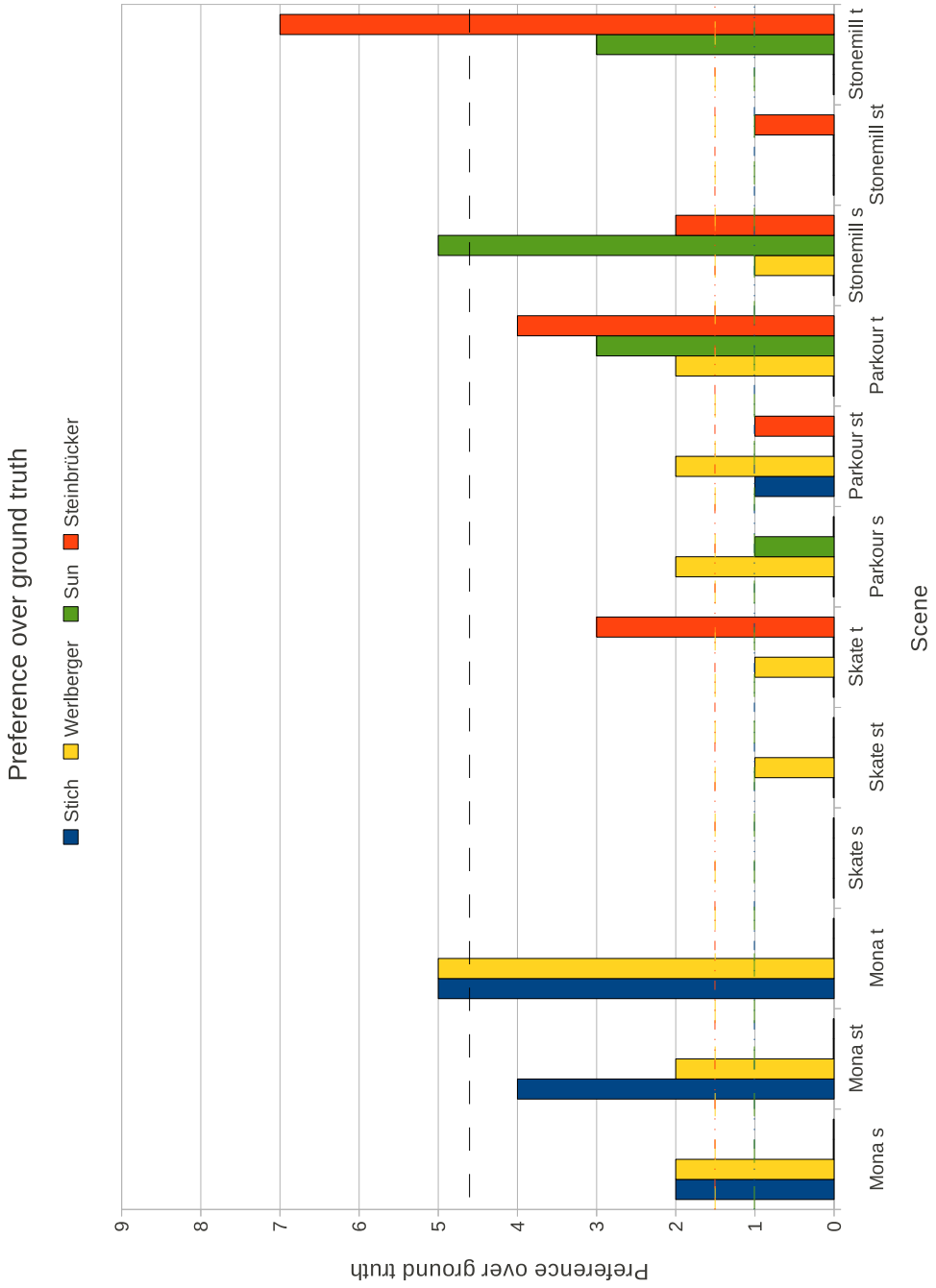


Figure 5.9: Preference of optical flow algorithms over ground truth flow fields, broken down by test scene. Values around 4.5 denote that both conditions are of equal perceived quality. None of the tested algorithms is able to reach this goal.

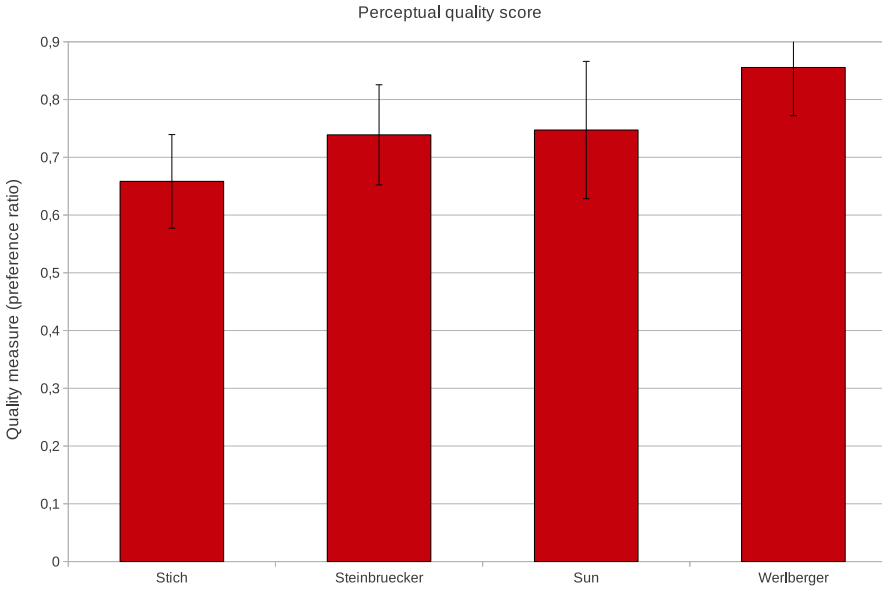


Figure 5.10: Perceptual quality scores leaving out comparisons against ground truth.

ground truth for the temporal pair. However, both algorithms are not able to perform equally well on the other two image pairs. The algorithms by Werlberger et al. and Stich et al. fail to maintain important structures of the building.

To address the second goal of our study, we repeat the evaluation without taking the comparisons against ground truth into account, Fig. 5.10. The insignificant one-way anova ($F(3, 32) = 5.354, p = 0.001$) indicates that there is no statistical difference between the individual approaches with respect to interpolation quality. Again, a Tukey test with significance level $p = 0.01$ confirms this.

5.5 Conclusion

To summarize, we have evaluated four recent optical flow algorithms in the context of image interpolation. We evaluated their perceptual quality on three typical real-world scenes used in the Virtual Video Camera system. A psychophysical user study has shown that no algorithm consistently reaches ground truth interpolation quality on every scene. The approach proposed by Stich et al. [140] performs well on scenes with a simple background and a few distinct edges, but fails as soon as the edge structure of the

Figure 5.11: Interpolation results on the temporal image pairs. For the Mona sequence (first column), the algorithms by Stich et al. [140] (fifth row) and Werlberger et al. [162] (fourth row) are on par with ground truth (first row). For the Skateboarder sequence, no algorithm is able to reach ground truth interpolation quality. For the Parkour sequence (third column), only the approach by Steinbrücker et al. [136] (third row) obtains a quality comparable to ground truth.

(a) Interpolation results on the temporal pairs using ground truth flow fields.

(b) Interpolation results on the temporal pairs using flow fields generated with Sun et al. [143].

(c) Interpolation results on the temporal pairs using flow fields generated with Steinbrücker et al. [136].

(d) Interpolation results on the temporal pairs using flow fields generated with Werlberger et al. [162].

(e) Interpolation results on the temporal pairs using flow fields generated with Stich et al. [140].

Figure 5.12: Interpolation results on the spatial image pairs. No algorithm reaches ground truth interpolation quality for any sequence.

(a) Interpolation results on the spatial pairs using ground truth flow fields.

(b) Interpolation results on the spatial pairs using flow fields generated with Sun et al. [143].

(c) Interpolation results on the spatial pairs using flow fields generated with Steinbrücker et al. [136].

(d) Interpolation results on the spatial pairs using flow fields generated with Werlberger et al. [162].

(e) Interpolation results on the spatial pairs using flow fields generated with Stich et al. [140].

Figure 5.13: Interpolation results on the spatio-temporal image pairs. Only on the Mona sequence (first column), the algorithm by Stich et al. [140] (fifth row) is able to reach a quality comparable to ground truth. On the Skateboarder and Parkour sequences, no algorithm reaches this goal.

(a) Interpolation results on the spatio-temporal pairs using ground truth flow fields.

(b) Interpolation results on the spatio-temporal pairs using flow fields generated with Sun et al. [143].

(c) Interpolation results on the spatio-temporal pairs using flow fields generated with Steinbrücker et al. [136].

(d) Interpolation results on the spatio-temporal pairs using flow fields generated with Werlberger et al. [162].

(e) Interpolation results on the spatio-temporal pairs using flow fields generated with Stich et al. [140].

images becomes too complex. The current top-performer on Middlebury with respect to the angular and endpoint error measures [143] only produces convincing results on the synthetic sequence, but produces noticeable artifacts on all real-world scenes. The approach by Werlberger et al. [162] produces good results on scenes with moderate motion and is at its most impressive in untextured regions. Similar to Stich et al. [140], it maintains important edge structures due to the anisotropic regularization. However, being based on a pyramidal approach, it fails to recover the motion of small, fast objects. In contrast to this, the long-range method by Steinbrücker et al. [136] is able to recover fast motion of small objects, even over large pixel distance. Unfortunately, it suffers from spurious wrong matches resulting from ambiguities in the global optimization approach. This most strongly shows up in disoccluded areas where all of the tested algorithms have problems.

As a conclusion, no algorithm is directly applicable for multi-view interpolation. The optimal algorithm would be a combination of a strong anisotropic regularization as proposed by Werlberger et al. [162] with a long-range, global optimization approach as proposed by Steinbrücker et al. [136]. We will pick up on this in Ch. 6 and show how to combine the advantages of the tested methods, extending them with a more expressive descriptor and a way to suitably treat disoccluded areas.

6

Symmetry and SIFT Descriptors for High-quality Long-range Flow

6.1 Introduction

In the previous chapter, we have evaluated the state-of-the-art in optical flow research for the use in multi-view interpolation algorithms. We have shown that none of the tested algorithms can be applied as is for this task. However, each of the tested algorithms has some particular strengths: the approach by Werlberger et al. [162] scores in untextured regions with a strong anisotropic regularization, Stich et al.'s algorithm maintains perceptually important image structures and allows for user interaction/correction, and the long-range correspondence estimation algorithm by Steinbrücker et al. [136] shows its strengths in scenes with fast motion and fine structures.

In this chapter, we combine the desirable properties of each of those algorithms in a common framework. In addition to this, we extend the formulation by a symmetry term to minimize ghosting effects originating in unsymmetric correspondence fields. Further on, enforcing symmetry in the flow fields helps to detect occlusion in a post-processing step. Since our test scenes also exhibit complex structures which often can't be distinguished on image brightness or color alone, we further propose to add more expressive descriptors to correspondence estimation. To this end, we integrate the SIFT descriptor [97] into the estimation process, leading to a more robust correspondence estimation. Our proposed symmetric long-range correspondence estimation algorithm is introduced in Ch. 6.2. This is followed by a performance evaluation on the test scenes

introduced in Ch. 5.4 in Ch. 6.3.

6.2 Correspondence Estimation

Our correspondence estimation algorithm is based on the approach presented by Steinbrücker et al. [136]. This approach separates the data-term, i.e. the brightness constancy assumption $I_1 - I_2(\mathbf{x} + \mathbf{w}_{1,2}) \approx 0$, and the smoothness-term, i.e. $\nabla \mathbf{w}_{1,2} \approx \vec{0}$, that are the basis for the estimation of the correspondence map $\mathbf{w}_{1,2}$. The key idea of this approach is based on the work of Zach et al. [175] where instead of direct minimization of the total-variation L^1 formulation

$$\min_{\mathbf{w}_{1,2}} \int_\Omega \alpha |I_1 - I_2(\mathbf{x} + \mathbf{w}_{1,2})| + |\nabla \mathbf{w}_{1,2}| \; d\mathbf{x}, \tag{6.1}$$

an auxiliary variable $\tilde{\mathbf{w}}_{1,2}$ is introduced and the problem

$$\min_{\mathbf{w}_{1,2}, \tilde{\mathbf{w}}_{1,2}} \int_\Omega \alpha |I_1 - I_2(\mathbf{x} + \mathbf{w}_{1,2})| + \frac{2}{\theta} \|\mathbf{w}_{1,2} - \tilde{\mathbf{w}}_{1,2}\|_2^2 + |\nabla \tilde{\mathbf{w}}_{1,2}| \; d\mathbf{x} \tag{6.2}$$

is considered. For small θ the solution of the original problem and the auxiliary problem are the same, but the latter problem permits an elegant and fast solution: Equation (6.2) is solved iteratively for $\tilde{\mathbf{w}}_{1,2}$ keeping $\mathbf{w}_{1,2}$ fixed, and for $\mathbf{w}_{1,2}$ keeping $\tilde{\mathbf{w}}_{1,2}$ fixed, see [175] for details. Depending only on $\mathbf{w}_{1,2}$ and no longer on $\nabla \mathbf{w}_{1,2}$ the latter problem can be solved point-wise by considering

$$\tilde{E}(\mathbf{x}) = \alpha |I_1 - I_2(\mathbf{x} + \mathbf{w}_{1,2})| + \frac{2}{\theta} \|\mathbf{w}_{1,2} - \tilde{\mathbf{w}}_{1,2}\|_2^2. \tag{6.3}$$

Since Eq. (6.3) can be solved point-wise, essentially any non-linear data-term can be used and optimized globally by an exhaustive search. This in particular allows for the integration of data terms that are not differentiable as has been shown in Ref. [137].

6.2.1 SIFT data term

We exploit this desirable property and additionally integrate the SIFT descriptor [97] into the data term. Our data term now reads

$$\begin{aligned} \tilde{E}(\mathbf{x}) &= \alpha \left(|I_1 - I_2(\mathbf{x} + \mathbf{w}_{1,2})| + \|S_1 - S_2(\mathbf{x} + \mathbf{w}_{1,2})\|_2 \right) \\ &+ \frac{2}{\theta} \|\mathbf{w}_{1,2} - \tilde{\mathbf{w}}_{1,2}\|_2^2, \end{aligned} \tag{6.4}$$

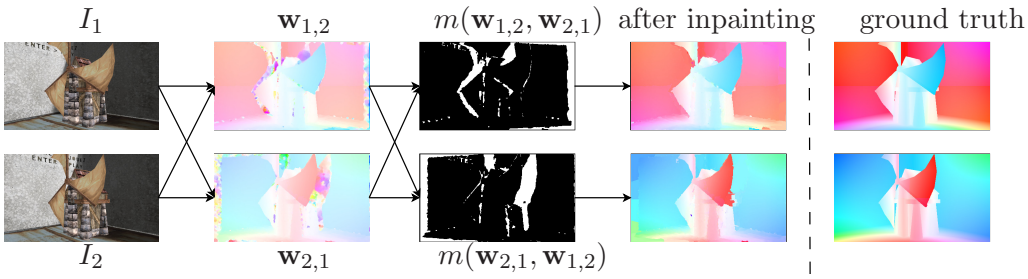


Figure 6.1: We first estimate correspondences by optimizing Eq. (6.6). We then detect occluded regions, compute color statistics along the boundary of the occluded regions and inpaint the flow values based on color similarity. The last column shows a color coding of the ground truth flow fields for visual comparison.

with S_1 and S_2 denoting the dense SIFT image for I_1 and I_2, respectively. An important implication of using the SIFT descriptor in the data term is that we are now restricted to integer valued flow vectors since interpolation between SIFT descriptors is not well defined. However, by incorporating the SIFT descriptor into the data term, we gain increased robustness against illumination changes, incorporating a small neighborhood into the data term. This idea of using more expressive descriptor for dense correspondence computation is not new. Liu et al. [96] proposed to use dense SIFT descriptors to match a query image to images in a large database and to extract the most similar ones.

6.2.2 Edge data term

Following the work of Stich et al. [140], maintaining edge correspondences is important for high-quality image interpolation. We thus further integrate the edge-matching approach into Eq. (6.4), yielding

$$\begin{aligned} \tilde{E}(\mathbf{x}) &= \alpha \left(|I_1 - I_2(\mathbf{x} + \mathbf{w}_{1,2})| + \|S_1 - S_2(\mathbf{x} + \mathbf{w}_{1,2})\|_2 \right) \\ &+ \beta f(E_1) \|E_1 - \mathbf{w}_{1,2}\|_2 \\ &+ \frac{2}{\theta} \|\mathbf{w}_{1,2} - \tilde{\mathbf{w}}_{1,2}\|_2^2, \end{aligned} \tag{6.5}$$

where E_1 is a sparse correspondence prior derived from matched edge pixels and $f(x) = 1$ iff E_1 has a valid entry and $f(x) = 0$ otherwise.

6.2.3 Symmetry data term

Symmetry is another important aspect for high-quality image interpolation. The input images are warped towards each other and the pixel values are blended; mismatching pixel values will lead to visible artifacts. Enforcing a symmetry constraint already in the computation of the flow fields is thus the basis for high-quality interpolation without cross-fading artifacts. We hence further add a symmetry term to Eq. (6.5), resulting in

$$\begin{aligned}\tilde{E}(\mathbf{x}) &= \alpha\left(|I_1 - I_2(\mathbf{x}+\mathbf{w}_{1,2})| + \|S_1 - S_2(\mathbf{x}+\mathbf{w}_{1,2})\|_2\right) \\ &+ \beta f(E_1)\|E_1 - \mathbf{w}_{1,2}\|_2 \\ &+ \gamma g(\|\mathbf{w}_{1,2} + \mathbf{w}_{2,1}(\mathbf{x}+\mathbf{w}_{1,2})\|_2) \\ &+ \frac{2}{\theta}\|\mathbf{w}_{1,2} - \tilde{\mathbf{w}}_{1,2}\|_2^2, \end{aligned} \tag{6.6}$$

where $g(x) = 1 - 1/(1 + kx^2)$, $k = 0.25$, is a weighting function used by Ince and Konrad [76] and originally proposed by Perona and Malik [119]. This weighting function penalizes small flow deviations and leaves large deviations untouched, allowing other data terms to take control when no symmetry can be established. Regions where sensible correspondences can't be established are potentially occluded.

Eq. (6.6) is optimized by a full search in an $n \times n$ window making use of the compute power of recent GPUs. n denotes the maximal flow length in pixels in x- and y-direction. The optimization is started with $\theta = 2 \cdot n$ and is run for 10 iterations with θ decreasing linearly to 0.01. After each iteration, the result is median filtered in a 5×5 neighborhood and 10 smoothing iterations are applied. A good starting point for the remaining parameters is given by $\alpha = 8.0$, $\beta = 4.0$ and $\gamma = 1.0$

A similar energy functional has recently been proposed by Lipski et al. [95]. In contrast to our approach, they use belief propagation to optimize the functional on the CPU. To make the optimization operable, they propose to evaluate the data terms once for the whole search space and then compress this data suitably for use in future iterations.

6.2.4 Occlusion detection

Since we enforce symmetric flow in the optimization, we can now use the geometric mismatch of forward flow $\mathbf{w}_{1,2}$ and backward flow $\mathbf{w}_{2,1}$ to detect occluded regions by

considering $m(\mathbf{w}_{1,2}, \mathbf{w}_{2,1}) = \|\mathbf{w}_{1,2} + \mathbf{w}_{2,1}(\mathbf{x} + \mathbf{w}_{1,2})\|_2^2$. Thresholding the geometric mismatch m gives a binary occlusion map O, regions where flow values are not symmetric and hence unreliable.

Repairing unreliable flow regions has recently been tackled by Berkels et al. [20]. Similar to our approach, they treat occluded regions in a post-process and propose a variational formulation for motion inpainting that is aware to edges in the original image. While this approach gives good results, its performance depends strongly on the parameters which unfortunately vary over the image domain. We take a parameter independent approach and repair the flow values in disoccluded regions by transferring the idea of geodesic matting [9] to motion inpainting. To this end, we make the assumption that the occluded region belongs either to foreground or background and its affiliation can be determined by color. As a first step, we identify all occluded blobs $\{b_i\}_{1\ldots N}$ in the binary occlusion map O by detecting connected components; large blobs are split perpendicular to their major axis. The splitting threshold is empirically set to 100 pixels. For every blob b_i, we then examine a five pixel wide boundary around the blob and determine two boundary clusters $\mathcal{F}_i$ and $\mathcal{B}_i$ by clustering unoccluded flow values using k-means, Fig. 6.2. Following the work of Bai and Sapiro [9], we then estimate a color probability density function (PDF) $Pr(x|\mathcal{F}_i)$ on the support $\Omega_{\mathcal{F}_i}$ of $\mathcal{F}_i$, and $Pr(x|\mathcal{B}_i)$ on the support $\Omega_{\mathcal{B}_i}$ of $\mathcal{B}_i$, respectively. Color PDFs are estimated in CIE L*a*b space via the fast kernel density estimation proposed by Yang et al. [173]. The likelihood for a pixel $x \in \Omega_{b_i}$ of color $\vec{c}_x$ to belong to the foreground is then given by

$$P_{\mathcal{F}}(\vec{c}_x) = \frac{Pr(x|\mathcal{F})}{Pr(x|\mathcal{F}) + Pr(x|\mathcal{B})}.$$

The same reasoning applies to the likelihood for background membership. In order to identify the splitting boundary in b_i, we then compute the weighted geodesic distance $d(x)$, $x \in \Omega_{b_i}$, for foreground and background PDF for each pixel $x \in \Omega_{b_i}$. $d(x)$ is the smallest integral of a weight function over all paths C connecting the border of the blob to x and can be written as

$$d(s_1, s_2) = \min_{C_{s_1,s_2}} \int_0^1 |w \cdot \dot{C}_{s_1,s_2}| dp. \tag{6.7}$$

$C_{s_1,s_2}(p)$ is a path connecting the pixels s_1 and s_2. The weights w are set to the gradient of the likelihood that a pixels belongs to the foreground respectively background. For

Figure 6.2: Flow repair: For each occluded blob, we determine two boundary classes $\mathcal{F}_i$ (green) and $\mathcal{B}_i$ (blue) by clustering the flow values along the boundary of the occluded blob b_i (red). For each boundary class, color statistics are computed in Lab space and each pixel of the occluded blob is inpainted with flow values from the most similar boundary.

a discrete grid of pixels, we can approximate Eq. (6.7) in a 4-point stencil by

$$\begin{aligned} d(s_1, s_2) &= \min_{C_{s_1,s_2}} \sum_{x,y} w_{x,y}, \\ w_{x,y} &= |P_{\mathcal{F}}(\vec{c}_x) - P_{\mathcal{F}}(\vec{c}_y)|, \quad x, y \in C_{s_1,s_2}. \end{aligned}$$

By the triangle inequality of distance functions, the geodesic distance guarantees that each border class results in exactly one connected component [9]. To inpaint flow values in the the blob b_i, we copy the median flow vector from the assigned border class. Using the median over mean, we gain increased robustness to outliers.

6.2.5 User interaction

Despite the expressiveness of the SIFT descriptor, there are cases where the automatic correspondence estimation does not produce correct results. To still guarantee high quality interpolation results, we resort to user interaction in difficult cases. User correspondences $\hat{\mathbf{w}}_{1,2}$ are specified in an interactive tool by drawing corresponding brush

strokes and are readily integrated into Eq. (6.6), resulting in

$$\begin{aligned}\tilde{E}(\mathbf{x}) &= \alpha\left(|I_1 - I_2(\mathbf{x}+\mathbf{w}_{1,2})| + \|S_1 - S_2(\mathbf{x}+\mathbf{w}_{1,2})\|_2\right)\\ &+ \beta f(E_1)\|E_1 - \mathbf{w}_{1,2}\|_2\\ &+ \gamma(1 - g(\|\mathbf{w}_{1,2} + \mathbf{w}_{2,1}(\mathbf{x}+\mathbf{w}_{1,2})\|_2))\\ &+ \delta f(\hat{\mathbf{w}}_{1,2})\|\hat{\mathbf{w}}_{1,2} - \mathbf{w}_{1,2}\|_2\\ &+ \frac{2}{\theta}\|\mathbf{w}_{1,2} - \tilde{\mathbf{w}}_{1,2}\|_2^2.\end{aligned}$$

We use the same function $f(\cdot)$ as used for the edge prior which is 1 if the user specified correspondence map has a valid entry, and 0 otherwise.

6.2.6 Regularization

While the original formulation of Steinbrücker et al. [136] used an isotropic L^1-norm as regularizer on the flow field, we follow the regularization strategy proposed Werlberger et al. [162]. They proposed to use an anisotropic, image-driven total variation regularization based on the Huber norm. Since their approach is also based on the dual formulation by Zach et al. [175], this regularization can be readily integrated. We refer the reader to [162] for details on the implementation. Using an anisotropic, image-driven regularization scheme helps to better align flow discontinuities with important image structures.

6.3 Evaluation and Discussion

6.3.1 Numerical evaluation

We evaluate our proposed correspondence estimation algorithm on the test scenes introduced in Ch. 5. In a first step, we numerically evaluate the influence of each proposed data term using the synthetic *Stonemill* sequence. The eight test conditions are listed in Table 6.1, condition 0 denotes the original approach of Steinbrücker et al. [136]. The color data term was always included with a weight of $\alpha = 8.0$. If non-zero, the remaining parameters were set to $\beta = 4.0$, $\gamma = 1.0$ and $\delta = 0.0$. The results for the three image pairs are listed in Tables 6.2, 6.3 and 6.4. As becomes evident from the numerical evaluation, our proposed data terms clearly improve the interpolation error and normalized interpolation error over the original approach. For the spatial and spatio-temporal

Data term	0	1	2	3	4	5	6	7
edge		x			x		x	x
symmetry			x		x	x		x
SIFT				x		x	x	x

Table 6.1: Test configurations of the proposed algorithm used for evaluating the contribution of each data term. An x denotes a non-zero weight for the respective data term.

pairs, our full approach with SIFT, symmetry and edge data term yields the lowest interpolation error. Furthermore, our algorithm has a better or similar interpolation error as the evaluated state-of-the art algorithms, cf. Table 5.2. However, looking at the average angular error and the endpoint error, our proposed extensions, especially SIFT and symmetry, lead to an increase of those errors. For the SIFT data term, this is due to the nature of the SIFT descriptor which incorporates a small neighborhood and has a poor localization in general. The symmetry data term tries to find the best compromise for forward and backward flow to be symmetric which also leads to an increase of the numerical errors. Only the edge data term is beneficial for improving these two error measures.

6.3.2 User study

To assess the perceived interpolation quality, we further carried out a user study similar to Ch. 5.4. Our participants compared the interpolation results produced from flow fields of our full approach, condition 7 in Table 6.1, to the interpolation results of all algorithms tested in Ch. 5.4. We again used a two-forced-alternative-choice task in which two interpolation sequences were shown successively and participants were asked to indicate which sequence contained less artifacts. For each of the four different test scenes, we compared all four different optical flow algorithms introduced in Ch. 5.4 as well as interpolations created from ground truth flow fields against our approach, yielding a total of $4 \cdot 5 \cdot 3 = 60$ trials. The perceptual user study was carried out the same manner as described in Ch. 5.4. Our test group consisted of 7 participants with strong computer graphics background.

For the analysis, we again determined a perceptual quality score for each algorithm by counting how many times it was chosen to produce fewer visual artifacts when compared to one of the other algorithms. The normalized scores are shown in Fig. 6.3

Case	Warp	AAE	AEE	IE	NIE
0	$\mathbf{w}_{1,2}$	10.45	3.00	35.96	3.65
	$\mathbf{w}_{2,1}$	14.01	4.73		
1	$\mathbf{w}_{1,2}$	7.51	2.52	32.83	3.47
	$\mathbf{w}_{2,1}$	10.74	4.22		
2	$\mathbf{w}_{1,2}$	8.58	4.24	30.65	3.11
	$\mathbf{w}_{2,1}$	10.43	5.37		
3	$\mathbf{w}_{1,2}$	16.39	5.00	30.82	2.98
	$\mathbf{w}_{2,1}$	19.06	7.04		
4	$\mathbf{w}_{1,2}$	8.19	3.87	30.56	3.07
	$\mathbf{w}_{2,1}$	10.34	5.04		
5	$\mathbf{w}_{1,2}$	9.04	3.31	28.51	2.81
	$\mathbf{w}_{2,1}$	13.52	5.25		
6	$\mathbf{w}_{1,2}$	16.03	4.91	30.88	3.00
	$\mathbf{w}_{2,1}$	18.39	6.80		
7	$\mathbf{w}_{1,2}$	8.92	3.30	28.39	2.80
	$\mathbf{w}_{2,1}$	13.24	5.14		

Table 6.2: Spatial image pair, stonemill sequence. Best values are marked in green.

Case	Warp	AAE	AEE	IE	NIE
0	$\mathbf{w}_{1,2}$	10.55	3.35	34.72	3.23
	$\mathbf{w}_{2,1}$	16.88	5.26		
1	$\mathbf{w}_{1,2}$	7.82	2.63	30.84	2.85
	$\mathbf{w}_{2,1}$	15.39	4.77		
2	$\mathbf{w}_{1,2}$	18.83	5.13	30.10	2.73
	$\mathbf{w}_{2,1}$	18.02	5.45		
3	$\mathbf{w}_{1,2}$	14.73	4.51	30.27	2.71
	$\mathbf{w}_{2,1}$	21.81	6.51		
4	$\mathbf{w}_{1,2}$	18.65	4.98	29.75	2.68
	$\mathbf{w}_{2,1}$	17.86	5.31		
5	$\mathbf{w}_{1,2}$	17.91	4.55	29.03	2.52
	$\mathbf{w}_{2,1}$	17.07	5.02		
6	$\mathbf{w}_{1,2}$	14.35	4.32	29.84	2.66
	$\mathbf{w}_{2,1}$	21.34	6.43		
7	$\mathbf{w}_{1,2}$	17.35	4.50	28.88	2.50
	$\mathbf{w}_{2,1}$	17.01	5.06		

Table 6.3: Spatio-temporal image pair, stonemill sequence. Best values are marked in green.

Case	Warp	AAE	AEE	IE	NIE
0	$\mathbf{w}_{1,2}$	2.98	0.64	8.84	0.58
	$\mathbf{w}_{2,1}$	3.22	0.66		
1	$\mathbf{w}_{1,2}$	3.12	0.64	7.62	0.51
	$\mathbf{w}_{2,1}$	3.01	0.63		
2	$\mathbf{w}_{1,2}$	8.58	0.97	7.80	0.61
	$\mathbf{w}_{2,1}$	8.33	0.95		
3	$\mathbf{w}_{1,2}$	4.30	0.71	7.51	0.56
	$\mathbf{w}_{2,1}$	4.36	0.71		
4	$\mathbf{w}_{1,2}$	5.81	0.78	7.42	0.54
	$\mathbf{w}_{2,1}$	5.68	0.76		
5	$\mathbf{w}_{1,2}$	4.56	0.79	7.58	0.55
	$\mathbf{w}_{2,1}$	4.41	0.78		
6	$\mathbf{w}_{1,2}$	4.17	0.71	7.44	0.55
	$\mathbf{w}_{2,1}$	4.22	0.71		
7	$\mathbf{w}_{1,2}$	7.30	0.71	7.37	0.54
	$\mathbf{w}_{2,1}$	7.29	0.83		

Table 6.4: Temporal image pair, stonemill sequence. Best values are marked in green.

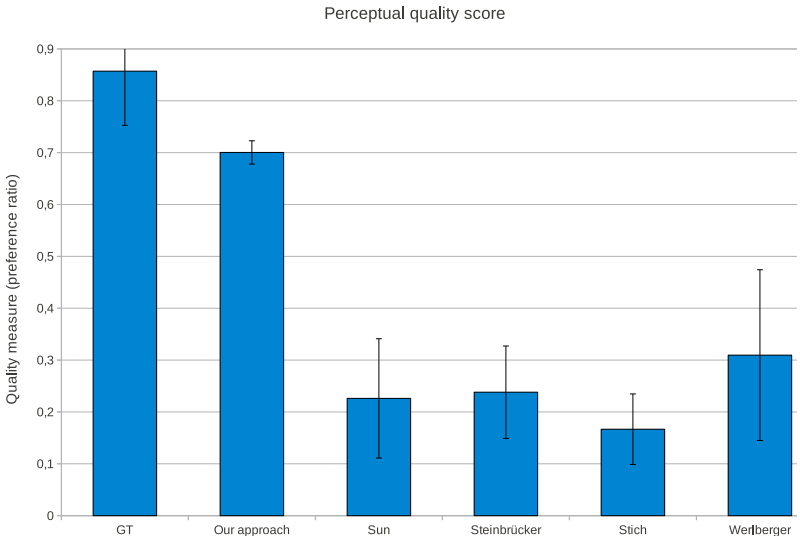


Figure 6.3: Perceptual quality scores for 6 different test conditions (optical flow algorithms).

for all six approaches. As can be seen immediately, there is a significant difference between perceived quality of interpolations based on flow fields of our algorithm and perceived quality of all interpolations generated from the other algorithms. This is confirmed by the highly significant one-way anova ($F(5,36)=54.409$, $p=0.001$). A Tukey test with confidence level $p = 0.05$ further reveals that our interpolations and ground truth interpolations differ significantly from the other interpolation results. While our approach still does not reach perceptual quality comparable to ground truth (there is a statistical difference confirmed by the Tukey test), the gap to ground truth clearly narrowed, see Fig. 5.8 for a comparison. Further, it can be observed that the inclusion of our approach into the user study shifted the preference of participants towards our approach which also gives a clear indication of better perceived quality. Leaving out ground truth comparisons, Fig. 6.4 shows how often our approach was preferred over any of the four other tested algorithms. Our algorithm was considered to be perceptually more plausible in 77% of the tests on average.

For visual assessment the ground truth interpolation, an interpolation produced from flow fields generated with the original algorithm of Steinbrücker et al. [136] and our results are shown in Figs. 6.5, 6.6 and 6.7. For all three test scenes, we obtain a higher perceptual interpolation quality than the original approach: correspondences over long

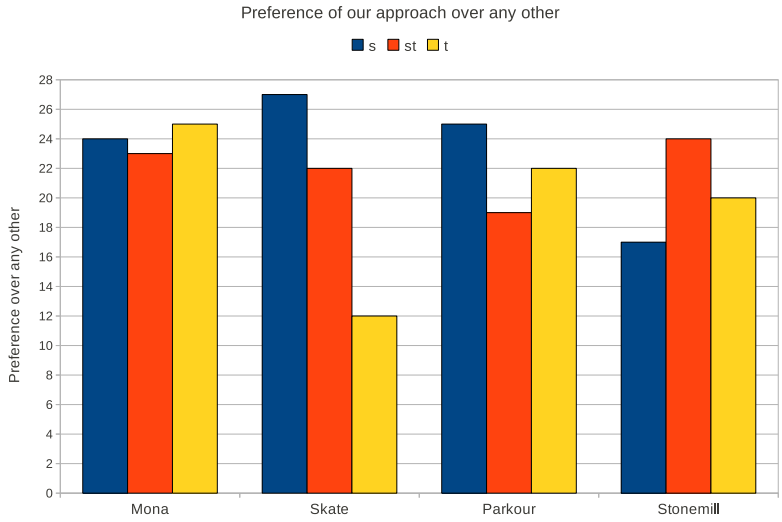


Figure 6.4: Preference of our algorithm over any of the other tested algorithms by scene and without taking comparisons to ground truth into account. The maximal number of votes per test case is 28. Our algorithm was preferred in 77% of all tests on average.

distances are estimated correctly and interpolation artifacts from wrong matches are greatly reduced. For the *Mona* sequence, our approach can hardly be distinguished from ground truth interpolations. Our algorithm is able to correctly estimate the motion of the background for the *Skateboard* sequence. Further on, also the untextured regions between the ramp and the background are mostly estimated correctly. Our approach still produces some artifacts in the foreground around the edge of the ramp, however, those errors could easily be corrected with a few brush strokes. The most complex *Parkour* sequence also poses a challenge for our algorithm and the resulting flow fields are not perfect. However, compared to the original approach and the results from top-performing algorithms, cf. Ch. 5.4, our approach again greatly reduces the estimation errors. While traditional variational approaches only manage to match the foreground regions correctly and produce an over-smoothed solution in the background, our algorithm also succeeds in matching parts of the cluttered background correctly.

Summarizing, SIFT and symmetry data terms are important to get long-range correspondences right and to stabilize the estimation in highly cluttered scenes such as the *Parkour* sequence. The global optimization strategy also helps to preserve fast motion and fine details. With respect to the numerical error evaluation, SIFT and

(a) Ground truth interpolation for the Mona scene. From left to right: spatial pair, spatio-temporal pair and temporal pair.

(b) Interpolation based on Steinbruecker for the Mona scene. From left to right: spatial pair, spatio-temporal pair and temporal pair.

(c) Interpolation based on our proposed approach for the Mona scene. From left to right: spatial pair, spatio-temporal pair and temporal pair.

Figure 6.5: Results on Mona scene.

(a) Ground truth interpolation for the Skateboard scene. From left to right: spatial pair, spatio-temporal pair and temporal pair.

(b) Interpolation based on Steinbruecker for the Skateboard scene. From left to right: spatial pair, spatio-temporal pair and temporal pair.

(c) Interpolation based on our proposed approach for the Skateboard scene. From left to right: spatial pair, spatio-temporal pair and temporal pair.

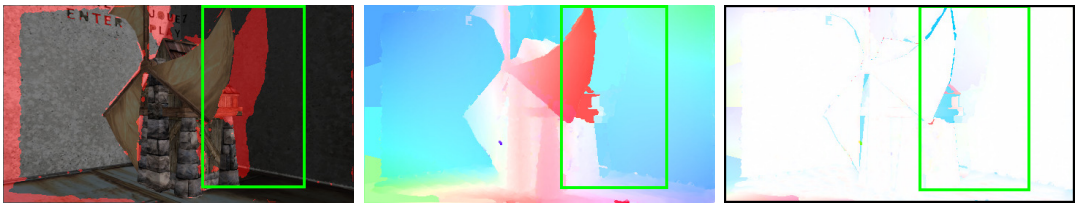
Figure 6.6: Results on Skateboard scene.

(a) Ground truth interpolation for the Parkour scene. From left to right: spatial pair, spatio-temporal pair and temporal pair.

(b) Interpolation based on Steinbruecker for the Parkour scene. From left to right: spatial pair, spatio-temporal pair and temporal pair.

(c) Interpolation based on our proposed approach for the Parkour scene. From left to right: spatial pair, spatio-temporal pair and temporal pair.

Figure 6.7: Results on Parkour scene.



(a) Source image overlaid with detected occluded regions. (b) Repaired flow field. Repair fails in the marked region. (c) Difference to ground truth flow field.

Figure 6.8: Flow repair fails if new entities are discovered or color statistics are not distinctive enough. In this case, the roof of the windmill is a new entity which violates all of our assumptions.

symmetry data term clearly improve the interpolation error but impair the angular and endpoint error measures. The edge data term, in contrast, helps to decrease the angular and endpoint error measures. Overall, our approach performs better on all our test scenes compared to state-of-the-art optical approaches.

6.3.3 Limitations

The inpainting approach used to fill occluded regions with sensible flow information relies on the assumption that the scene consists of two layers and that the occluded region is in direct adjacency to the region it actually belongs to. We further make the assumption that the layers can be clearly distinguished by their color statistics. If one of those assumptions is violated, as is the case for parts of the roof in the windmill sequence, Fig. 6.8, wrong flow information will be filled in. Incorporating high-level scene segmentation into the inpainting step might be an interesting direction for future work.

Despite strong parallelization and running the estimation on recent graphics hardware using CUDA, our approach suffers from a high computational cost. This running time is due to the nature of the optimization approach which is in the order of $O(n^2 \cdot w \cdot h)$, where n denotes the search window size and w and h denote the image resolution. This is further impaired by the comparison of 128-dimensional SIFT descriptors instead of 3-dimensional colors. Adding the SIFT term to the optimization approximately doubles the running time of the estimation, cf. Table 6.5. This is due to the fact that each descriptor has to be fetched from texture memory over and over

Scene	Resolution [pixel]	Search window [pixel]	Estimation [s]	Estimation w/o SIFT [s]
Stonemill	480×270	60	1330	638
Mona	960×540	90	13560	7720
Skateboard	960×540	180	48500	24715
Parkour	960×540	170	47800	22980

Table 6.5: Timing results of the proposed approach on the spatio-temporal image pairs of our test scenes. Timings are given for 10 iterations.

again since caching them in shared memory is impossible due to cache size restrictions. Efficient parallelization on recent GPUs is thus impossible. The estimation time for the bidirectional correspondence fields on the spatio-temporal pairs of our test scenes with and without SIFT descriptors is given in Table 6.5.

6.4 Conclusion

In this chapter, we have presented a robust method for long-range correspondence estimation based on SIFT, symmetry and edge data terms with an explicit occlusion reasoning in a post-process. We numerically evaluated the contribution of the individual data terms to four numerical error measures. Further on, we showed in a perceptual user study that our approach yields superior interpolations on our real-world test scenes.

In the next chapter, we will exploit the symmetry of the computed flow fields to derive a novel formulation of multi-image interpolation.

7

Label-based Multi-image Interpolation

7.1 Introduction

So far, this thesis investigated and discussed different approaches to image-based correspondence estimation in the context of image interpolation. These correspondences maps are the first vital ingredient to high-quality view synthesis. We now turn to the second crucial part in this context, which is the view synthesis algorithm itself.

The synthesis of in-between images from different viewpoints and/or time instants is experiencing a renaissance. Stich et al.[139, 140] recently introduced a perception motivated spatio-temporal image interpolation technique. Their approach is based on blending four forward-warped images with spatially varying blending weights. While this approach delivers high-quality interpolation results, it can suffer from ghosting and blurring artifacts as soon as the underlying correspondence fields are imperfect. Further, blending constitutes a low-pass filtering operation and the interpolated image will thus lose some high-frequency details.

The problem of ghosting artifacts stemming from a combination of image blending and incorrect correspondences has recently also been noticed by Mahajan et al. [99]. They tackled the problem by using a path-based image interpolation technique which avoids blending pixels. Their idea is that each pixel traces out a path in the original images during the transition from one image to the other. Along this path, they search for a point where both images are in good correspondence and then immediately

transition to the other image instead of blending two pixel values. The strength of this approach is hence that each pixel in the interpolated view is sampled from exactly one source image, thus avoiding ghosting or blurring artifacts. A major drawback of this approach however is that the path idea can only be applied to two images; a direct extension to multi-image interpolation without resorting to intermediate interpolated images is not feasible.

In this chapter, we combine the strengths of both approaches and propose a direct multi-image interpolation that avoids blending several images. Instead, in our approach we decide for each pixel in an interpolated sequence from which of the source images to sample best. Combined with our correspondence estimation technique presented in Ch. 6, our approach yields high-quality image interpolation results without ghosting.

The remainder of this chapter is structured as follows: we discuss related work in Ch. 7.2 before we describe our multi-image interpolation algorithm in Ch. 7.3. We evaluate the algorithm on several test scenes in Ch.7.4 and conclude with a brief discussion, Ch. 7.5

7.2 Related work

Starting with a generally applicable, feature-based method for interpolating between two different images in Ref. [17], image morphing techniques have a long history. In their seminal work, Chen and Williams [38] show how general image morphing can be used for view interpolation. For improved rendering performance, McMillan and Bishop [108] propose a planar-to-planar, forward mapped image warping algorithm. Mark et al. [101] adapt this method to achieve high frame rates for post-rendering, while Zhang et al. [176] apply feature-based morphing to light fields. Lee et al. extended the feature-based method presented by Beier and Neely [17] to more than two images [82]. Stich et al. [139; 140] recently proposed an algorithm for perceptually plausible image interpolation in space as well as time. This approach is the basis for real-time view synthesis in the Virtual Video Camera System [94], Ch. 3.5.3, suitably extended to more than two images. Recently, Mahajan et al. presented a path-based interpolation for image pairs that operates in the gradient domain and prevents ghosting/blurring and many occlusion artifacts visible in morphing-based methods [99].

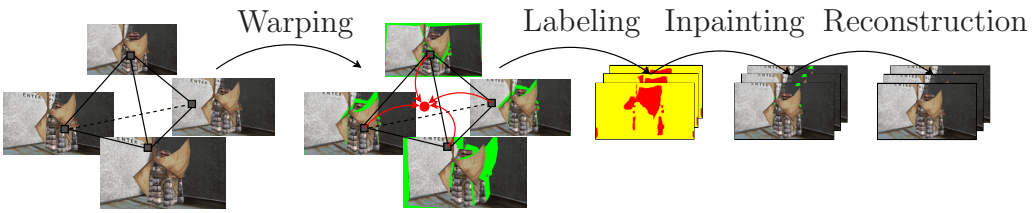


Figure 7.1: Schematic overview of the proposed interpolation. First, the input images are forward-warped to each target position of the interpolated sequence. The interpolated sequence is then constructed as the solution of a spatio-temporal labeling problem. Holes in the interpolated sequence are inpainted and the result is finally reconstructed in the gradient domain.

7.3 Multi-image interpolation

Stich et al. [139] proposed a GPU-based image interpolation algorithm that is founded on mesh-based forward warping and adaptive blending of images. In this approach, occlusion and disocclusion are handled by depth heuristics and a connectedness measure, respectively. This algorithm naturally extends to more than two images; the virtual view I_v can be synthesized as

$$I_v = \sum_{i=1}^{n} \mu_i \tilde{I}_i,$$

with

$$\tilde{I}_i \left(\mathbf{x} + \sum_{j=1,\ldots,n,j\neq i} \mu_j \mathbf{w}_{ij}(\mathbf{x}) \right) = I_i(\mathbf{x}) \tag{7.1}$$

and μ_i denoting a barycentric weighting scheme in $n-1$ dimensional space. The construction of the space and the derivation of the weights has already been described in Ch. 3. While this approach produces good results if the correspondence fields $\mathbf{w}_{ij}$ match up exactly, the blending actually produces a low-pass filtered image for imprecise correspondence fields. Further, the adaptive blending weights derived from the connectedness measure often result in streaking artifacts in disoccluded regions, cf. Fig. 7.2(a).

Our proposed approach also relies on forward warped images, i.e. we also warp each input image to the desired position by applying Eq. (7.1). In contrast to Stich et al., our approach cuts the underlying warping mesh open in disoccluded regions by measuring the triangle stretch. Instead of blending the forward warped images to get

the interpolated image, we formulate the interpolation as a labeling problem which is explained in the following subsection.

7.3.1 Graph-cut based interpolation

Inspired by the interpolation approach of Mahajan et al. [99], we now show how we avoid blending several images at each pixel. Our approach is based on solving an optimization problem that decides for each pixel in the virtual view I_v from which of the n source images best to take the color information. To this end, we formulate the view synthesis as a labeling problem in a 3D MRF framework incorporating temporal coherence. Note that from now on we consider the entire interpolated sequence and perform all computations on this spatio-temporal volume. The goal is to assign to each pixel $p \in I_v$ a label $L(p)$ indicating which of the source images $\tilde{I}_{L(i)}$ the pixel should be taken from. In particular, we optimize the following energy:

$$E(L) = \sum_{p \in I_v} E_D(p, L(p)) + \lambda \sum_{p,q \in \mathcal{N}} E_S(p, q, L(p), L(q)), \tag{7.2}$$

where E_D measures the quality of the current labeling, E_S controls the smoothness of the labeling and p, q are neighboring pixels in a 6-connected[1] neighborhood $\mathcal{N} \subset \{I_v\}$.

Our data cost function

$$E_D(p, L(p)) = 4.0 \cdot P_{dis}(\Delta_p) \cdot e^{1-\mu_{L(p)}},$$

favors pixel that receive a low disocclusion penalty $P_{dis}(\Delta_p)$. We compute the disocclusion penalty based on the area of the associated triangle Δ_p in the underlying warp mesh as

$$P_{dis}(\Delta_p) = \begin{cases} 0 & \text{if} \quad \Delta_p \leq 0.5 \\ e^{1.25(\Delta_p - 0.5)^2} - 1 & \text{else.} \end{cases}$$

We further assume that images with a high barycentric weight $\mu_{L(p)}$ only have a low distortion.

The smoothness term

$$E_S(p, q, L(p), L(q)) = X \cdot Y$$

is composed of

$$X = \left(\|\tilde{I}_{L(p)}(p) - \tilde{I}_{L(q)}(p)\|_2 + \|\tilde{I}_{L(p)}(q) - \tilde{I}_{L(q)}(q)\|_2 \right)$$

[1] Our neighborhood consists of 4 spatial and 2 temporal neighbors.

and

$$Y = 2.0 - (\nabla_{p,q}\tilde{I}_{L(p)} + \nabla_{p,q}\tilde{I}_{L(q)}),$$

see Bhat et al. [23]. It prefers cuts through regions of homogeneous colors or along prominent edge structures. When considered from a perception point of view, this is desirable since (1) cuts in homogeneous regions will most likely go unnoticed and (2) cuts along prominent edges will keep structural information intact.

We find a labeling that is the approximate global minimum of Eq. (7.2) using the alpha-expansion algorithm proposed by Boykov et al. [28]. For our test scenes, we set $\lambda = 4$.

7.3.2 Spatio-temporal inpainting

The forward warping approach with mesh cutting discussed in Ch. 7.3 introduces holes in each source image where pixels are disoccluded. Eventually, some of those areas are invisible in all source images, such that our algorithm does not find a sensible labeling for those areas and hence no color value, cf. Fig. 7.2(b). Those areas have to be filled with perceptually plausible color values in a temporally consistent manner. To this end, we adapt the inpainting method presented by Telea [145] to three dimensions and inpaint the spatio-temporal holes in the interpolated sequence. In our implementation, we favor inpainting along the temporal direction, over inpainting along the spatial dimensions by giving the temporal dimension a higher weight. This is justified as follows: invisible regions potentially occur at occlusion edges; inpainting along the spatial dimensions would lead to diffusing wrong color information over those occlusion boundaries. When inpainting along the temporal dimension, we exploit that the invisible region becomes visible at some point earlier or later in the sequence and we are hence able to diffuse color information in a perceptually plausible way, Fig. 7.2(c). We apply a temporal weighting factor of 0.9 and a spatial weighting factor of 0.1 in all our test scenes.

7.3.3 Label-based view synthesis

The labeling found by our optimization completely defines how to construct the virtual view I_v: one simply samples each interpolated pixel from the appropriate source

(a) Blending-based interpolation (b) Graph-cut-based interpolation without inpainting (c) Result of our complete algorithm

Figure 7.2: We evaluate the correctness of our algorithm on the Stonemill scene with ground-truth correspondence fields. Each image is interpolated from four input images. Our approach is superior to the blending-based approach, especially in disoccluded regions. In those regions, the blending-based approach suffers from annoying streaks, despite the use of ground truth flow fields.

image. We follow the approach of Mahajan et al. [99] and sample the image in the gradient domain. The interpolated image is then reconstructed by solving the 3D Poisson equation, i.e. we solve

$$\nabla^2 I_v = \nabla \cdot G,$$

where $G(\mathbf{x}) = (G_x, G_y, G_t)$ denote the gradients of the virtual view in the x and y direction and along the path through the spatio-temporal volume. The Laplacian operator is computed as $\nabla^2 I_v = \frac{\partial I_v}{\partial x^2} + \frac{\partial I_v}{\partial y^2} + \frac{\partial I_v}{\partial t^2}$ and the divergence of the gradient field is computed as $\nabla \cdot G = (\frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y} + \frac{\partial G_t}{\partial t})$. We take the first and last frame of the interpolation path as well as the one-pixel boundary of each intermediate frame as boundary conditions for the Poisson reconstruction.

7.4 Results and Evaluation

To evaluate the correctness of the proposed multi-image interpolation approach, we use the synthetic *Stonemill* sequence with ground-truth flow fields. We compare our proposed approach to the multi-image interpolation method proposed by Stich et al. [139], Fig. 7.2. Each image is interpolated from four input images. The blending-based interpolation technique suffers from streaking artifacts in disoccluded and totally invisible regions, despite the use of ground-truth correspondence fields. Our approach clearly produces less artifacts and is visually more pleasing. We next evaluate our interpolation approach on a high-speed camera sequence from the Middlebury optical

(a) Result of Mahajan et al. [99]

(b) Our result without symmetry and SIFT for correspondence estimation

(c) Our result with symmetry and SIFT for correspondence estimation.

Figure 7.3: Results on the backyard sequence from the Middlebury benchmark, interpolated from two images. Left: result of the Moving Gradients approach by Mahajan et al. [99], middle: our proposed interpolation using the approach of Steinbrücker et al. [136] for correspondence estimation, and right: our interpolation using the correspondence estimation approach presented in Ch. 6. The full approach yields comparable results to Mahajan et al.. Note that the left image has been extracted from video; the low quality might be due to low resolution.

(a) Result of Stich et al. [140] without user correction.

(b) Result of our complete algorithm.

Figure 7.4: Results on the Heidelberg stereo sequence, interpolated from three images. Left: result of the approach of Stich et al. [140] using their edge-based correspondence estimation. No user corrections have been applied. The image suffers from ghosting noticeable around the windows, and appears blurry overall. Right: our algorithm yields crisp images without ghosting.

flow benchmark. This scene features fast motion of small objects and represents a difficult test case for common optical flow approaches. We evaluate our interpolation algorithm on flow fields computed using the original algorithm of Steinbrücker et al. [136], and on flow fields computed using our algorithm proposed in Ch. 6. Without the extensions proposed in Ch. 6, the interpolation suffers from artifacts such as a distorted ball and visible seams running over the girl's skirt. Compared to the approach of Mahajan et al., we obtain visually comparable results with our full approach, Fig. 7.3. In addition to Mahajan et al., our approach naturally extends to more than two images without resorting to intermediate interpolations, thus avoiding additional image resampling that potentially leads to loss in quality. We next compare our full approach to the multi-scale approach recently proposed by Stich et al. [140], Fig. 7.4. Again, our approach produces sharper images without ghosting artifacts.

7.4.1 Limitations

Our algorithm can only be applied for off-line rendering. This is due to the fact that the graph-cut optimization as well as the spatio-temporal inpainting make use of previous and future frames. The spatio-temporal inpainting approach is further limited to regions with only little texture. In highly textured regions, our inpainting will not be able to fill in matching details. For future work, the examination of texture synthesis algorithms for this task could be a promising avenue.

7.5 Conclusion

In this chapter, we have presented an algorithm for ghosting-free multi-image interpolation. Our approach is based on a novel formulation of image interpolation as a labeling problem. Combined with our symmetric long-range correspondence estimation technique presented in Ch. 6, our interpolation technique yields high-quality results superior to state-of-the-art.

8

Space-Time Visual Effects

8.1 Introduction

In Ch. 3, we discussed the Virtual Video Camera, a purely image-based system for free-viewpoint navigation [94]. In this chapter, we apply this system for the task of space-time visual effects creation in motion picture and TV commercial production.

Visual effects can be broadly categorized by way of production: while many effects are based on traditional 3D computer graphics technology which are created off-line, space-time visual effects, referred to in the following as STF/X, are image-based and currently need to be recorded directly on-set. Since the release of the movie "The Matrix" [8], space-time visual effects have been used in a number of other motion picture and TV commercials [48]. To create these effects, time-slice photography and other special recording setups are being used Digital Air Inc. [48] to capture a real-world, dynamic scene in some unconventional way, e.g., to create *freeze-rotate*, *slow motion*, *motion blur*, *motion distortion* or *multi-exposure* effects. Typically, each frame of the desired effect sequence is recorded by a separate camera. On set, innumerous cameras must be exactly positioned and aligned, and shutter timings of all cameras must be precisely triggered. Once captured, the only way to alter a recorded effect is to re-take the entire shot, which is why a lot of time and effort has to go into planning and capturing each space-time visual effect. In summary, the contemporary production process of STF/X requires a lot of time, money and expensive hardware.

With a system as the Virtual Video Camera [94], we can overcome those limitations in STF/X production. We propose an alternative approach that allows creating space-

time visual effects as a post-production process. In separating STF/X creation from image acquisition, arbitrary visual effects can be interactively designed, edited and combined from the same recorded footage.

Our image-based approach makes use of the Virtual Video Camera System introduced in Ch. 3. Various space-time visual effects like *freeze-rotate shots*, *time ramps*, *space ramps*, *slow motion* and *match cut* Digital Air Inc. [48] are created by simply specifying the camera path through navigation space. By incorporating frame accumulation rendering, our STF/X editor also features camera shutter effects like *long exposure*, *multi-exposure* and *flash effects*. As our approach relies on image data only, artistic stylization techniques such as *speed lines* and *particle effects* can be applied as well. Finally, different space-time effects can be combined in arbitrary fashion. The resulting STF/X sequence can be viewed instantly at real-time frame rates for interactive refinement and editing.

After briefly discussing related work in the following section, we describe how different space-time visual effects can be realized in Ch. 8.3. Results for a selection of effects and a number of different real-world scenes are presented in Ch. 8.4 where we also discuss remaining challenges and limitations of our approach.

8.2 Related Work

In [168], Wolf gives an overview of various space-time visual effects for both still and video cameras. He introduces an insightful visualization scheme for space-time effects by specifying plenoptic function samples in two-dimensional space-time diagrams, Fig 8.1. Digital Air Inc. [48], a company specialized in image-based captured visual effects creation, uses specialized camera hardware to record these plenoptic samples directly from real-world scenes. On set, camera positions and directions, shutter settings and exposure times must all correspond precisely to the pre-planned plenoptic samples. Instead of recording specific plenoptic function samples, our approach is to use image-based rendering to synthesize samples of the plenoptic function from arbitrary, sparse sample recordings.

This idea is not completely new: Zitnick et al. [181] already presented some visual effects created with their system. Being limited to view interpolation alone, however, only a small subset of STF/X effects could be generated. Recently, Zheng et al. [177]

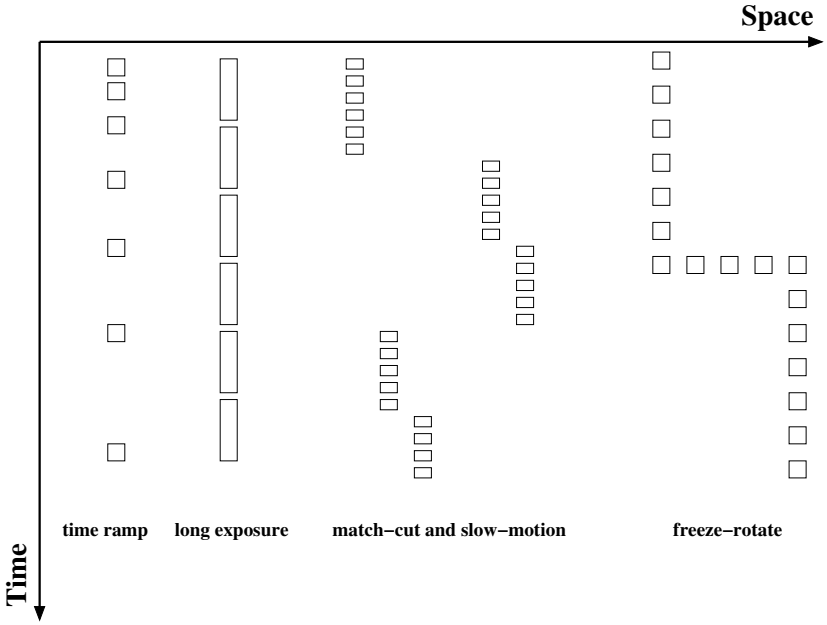
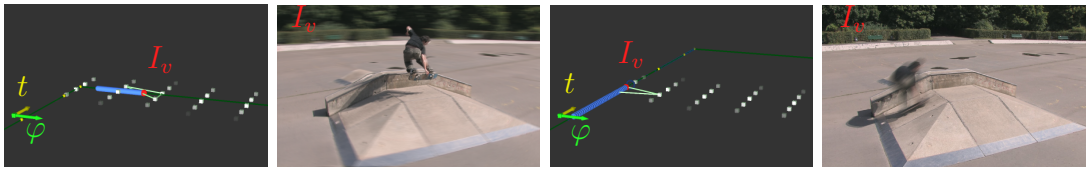


Figure 8.1: Spatio-temporal notation for various types of cinematic shots as introduced by Wolf [168]. Each rectangle denotes a camera frame, the elongation of the rectangle encodes the exposure time.

described a system for effect choreography within the confines of a static light field. Their system allows to create 3D pan & scan effects, but mainly targets still images and a small range of viewpoints instead of video footage. All of those approaches require some sort of geometry, either in the form of dense per-frame depth maps or a geometry proxy. Their applicability for STF/X production is hence limited by acquisition and/or 3D scene reconstruction constraints.

8.3 The STF/X Editor

The main contribution of this chapter is the STF/X editor, an extension of the Virtual Video Camera system discussed in Ch. 3. On the website of Digital Air Inc. [48] a list of visual effects are described that can be produced with specific camera setups and different exposure techniques. We distinguish between effects that are solely based on camera placement and different triggering schemes (*frozen moment, start-stop, slow motion, time ramps, space ramps and match cut*) and effects that rely on a combination of camera placement and exposure techniques (*space blur, time blur, multi-exposure,*



(a) Space blur sampling and resulting blurred scene background.

(b) Time blur sampling and resulting blurred skater.

Figure 8.2: Different space-time sampling examples and resulting images.

flash trail, open flash and progressive motion distortion)[168].

In the following, we show that these effects can be realized by our STF/X editor only from a single recording, shifting space-time visual effects creation entirely to the post-production stage.

8.3.1 Path-based effects

In [168], Wolf describes space-time visual effects in two-dimensional space-time diagrams, Fig. 8.1, allowing for intuitive visualization and planning of visual effects. In his representation, each effect maps to a specific path within the diagram. The navigation space of the Virtual Video Camera, cf. Ch. 3.3, is the three-dimensional counterpart of Wolf's space-time diagrams which can be interactively explored. Within this space, arbitrary camera paths are defined by placing, editing, or deleting control points making use of common 3D animation concepts. The effect camera path through space-time is interpolated by Catmull-Rom splines [36]. This way, various effects can be generated by designing an appropriate camera path: *frozen moments* map to paths with constant time, *slow motion* and *time/space ramps* are created by varying the sampling density along the path, and *match cuts* amount to jumps to specific points in space-time.

In contrast to contemporary space-time visual effects production, our approach is entirely a post-production process. We record a scene once with up to 16 cameras. Afterwards, we can create various space-time visual effects from this single capture as opposed to designing and capturing each single effect separately.

8.3.2 Exposure effects

Visual effects based on unconventional exposure settings are very common in still photography. The longer the shutter is open, the more light is captured over time, and

(a) Scanline progressive motion distortion.

(b) Particle effect.

(c) Speedline effect.

Figure 8.3: Shutter effect and motion stylization techniques.

moving objects appear blurred. When moving light sources in such a setup, a trail of light is captured, also known as "light painting". Using flash lights while exposing the film for a longer period of time can also create interesting effects. A single flash during a long-exposed shot shows a very short moment clearly visible and detailed while all other movements will be blurred.

Opening and closing the camera's shutter several times creates a composition of frames of the same scene at different times, called multi-exposure. This way, it is possible to display an object in motion in a single frame. In video production, however, these effects are created in post-production due to physical limitations of the longest possible exposure time. Instead of taking a single long exposure, the scene is typically captured with high frame-rate cameras with short exposure times, and the long exposure is synthesized via image accumulation.

Simulating an open shutter. To simulate an open camera shutter in our STF/X editor, we integrate along a parametric path $p(s)$, $s \in [0, 1]$, in the navigation space $\mathcal{N}$ of the Virtual Video Camera by accumulating virtual views. The resulting image is

$$\mathbf{I}_O = \int_{p(s)} I_v(p(s))\mathrm{d}s, \tag{8.1}$$

where $p(s)$ defines a sample position in $(\varphi, \theta, t) \in \mathcal{N}$. The virtual view I_v is synthesized according to Eq. (3.1). Suitably discretized, Eq. (8.1) serves as basis for various shutter effects.

Space blur, time blur and multi-exposure. Motion blur is the most prominent effect of long exposure times and has been thoroughly investigated. Motion blur render-

ing either relies on sampling the shutter interval, e.g. [4, 34, 67] or on the construction of an appropriate filter in space-time [49]. Image-space solutions create blur based on the motion field at a single instant [29, 106, 120]. We adapt the approach by Haeberli et al. [67] since it seamlessly integrates with flash effects described below.

Depending on the desired effect, i.e., blur in time, space or both, different sampling paths $p(s)$ have to be used to solve the integral in Eq. (8.1). A pure time blur, for example, can be created by sampling only along the temporal dimension, so $p(s)$ is parallel to the temporal axis of the navigation space. The discretized and weighted version of Eq. (8.1) then reads

$$I_O = \sum_{n=0}^{N-1} w(n) \cdot I_v(\varphi, \theta, t_0 + n \cdot \frac{(t_1 - t_0)}{N-1}), \tag{8.2}$$

with the weighting term $w(n), \sum_{n=0}^{N-1} w(n) = 1$. N describes the number of samples used in this temporal super-sampling strategy and steers the quality of the long exposure, with more samples resulting in higher quality. Sampling along one or both spatial axes of the navigation space $\mathcal{N}$ while keeping the time constant will result in space blur. Fig. 8.2 shows different sampling patterns in space-time for time and space blur along with the resulting image. By reducing the number of samples N, the motion blur is reduced, resulting in a composition of N distinct moments in space-time within one frame, called multi-exposure. An example is shown in Fig. 8.5(d). For the multi-exposure effect, all samples are weighted equally and the sampling pattern moves through space-time with the camera position.

Flash effects. Combining long exposures with a single or multiple flash strobes, *open flash* and *flash trail* effects are generated. Again, using the accumulation technique of Eq. (8.2), these effects are generated by combining a long exposure with one or more short exposure renderings taken at the respective point in space-time. The weights are adjusted such that higher weight is given to the flash samples. In contrast to the multi-exposure technique, the flash samples remain fixed in space-time, resulting in a trail of exposures through space-time for moving objects. An example of *open flash* is shown in Fig. 8.5(a).

8.3.3 Advanced shutter effects

Our approach is not restricted to effects based on exposure techniques. We are also able to simulate arbitrary - and physically impossible - shutter timings. Consider for example a rolling shutter: such a shutter often results in a progressive distortion of a moving object where each scanline depicts a slightly different instant of time, Fig 8.3(a). To emulate this effect in our STF/X editor, the rendered special effect is composed of scan lines taken from different samples along the temporal axis in space-time. We are not restricted to sampling only along the temporal axis; sampling along the spatial axes would result in cubistic views of an object. In principle, each pixel of the virtual view can be taken from any arbitrary position in our space-time navigation space.

8.3.4 Non-photorealistic effects

Our system is not limited to photorealistic effects. Since we are operating on images only, any artistic stylization that can be applied to a single image or a video stream naturally maps to our STF/X editor as well. There is a vast amount of literature on different aspects of artistic image stylization, from cartoon effects [155] and video abstraction [166], water color and painterly rendition [26, 70] to video-based illustration of motion [78]. We extend the algorithms for motion stylization proposed by Kim and Essa [78] and integrate them into our STF/X editor. Motion stylization is typically applied to objects to amplify their perceived motion. In contrast to [78], we do not segment the moving object on-the-fly but use the binary masks created in a preprocessing stage. At the current stage, we use commercial software [2] for this purpose and process each video stream accordingly. However, alternative approaches such as Video SnapCut [10] could be employed as well. In order to get a mask for an arbitrary point in space-time, we apply the same rendering technique as for the virtual views described in Ch. 3.5.3. We then use those masks to track a set of particles along the trailing contour of the object. These particle tracks serve as basis for the rendition of particles or speedlines as shown in Figs. 8.3(b),(c).

8.3.5 The effect graph

Since all of the described effects use only image data as input to render F/X images as output, it is straight-forward to combine effects. A multi-stage rendering pipeline

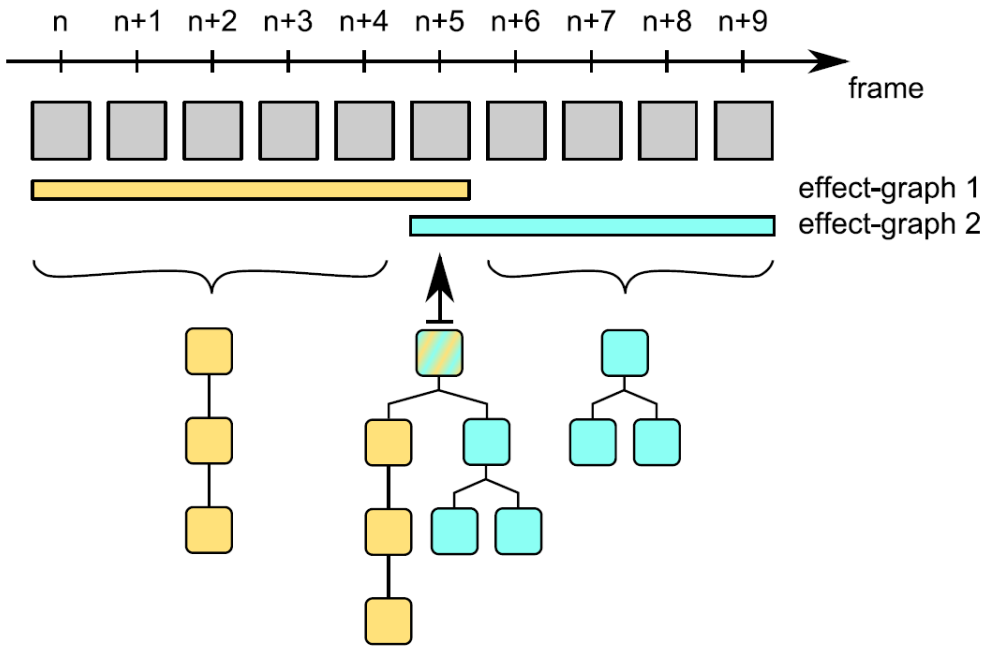


Figure 8.4: Effects can be combined in a digraph structure to create the final output frame. Effect graphs are allowed to overlap, resulting in a weighted combination of the overlapping subgraphs.

can be setup to create even more complex visual effects. To define a proper rendering flow, effects are combined in a digraph structure. The root of the graph holds the effect responsible for rendering the final frame. All other vertices in the graph act as input sources for connected effects, with directed graph edges indicating the flow of image data. Effects with no input sources take the underlying STF/X editor as data source. The edges of the graph can be weighted, allowing emphasis of one effect over another, Fig. 8.4. An example of such an effect combination is shown in Fig. 8.5(a) where we combine *space blur* with *open flash*.

8.3.6 Implementation details

Many of the effects described above require a larger number of space-time samples as input. While the underlying STF/X editor allows interaction in real-time, performance quickly deteriorates if many space-time samples have to be rendered over and over again. Fortunately, it is very likely that space-time samples can be reused when rendering many

consecutive frames with the same visual effect. This especially holds true for effects based on accumulation. We exploit this property by incorporating a least-recently-used caching strategy [116] into our STF/X editor, caching previously used space-time samples. This way, the rendering overhead is reduced to a minimum and the real-time performance of the system is kept.

8.4 Results and Discussion

We evaluate our STF/X editor on the *Skateboarder* and *Firebreather* sequences. The challenges that come with the *Skateboarder* sequence have already been discussed in Ch. 5.4, the *Firebreather* sequence features high dynamic contrast, over-exposed areas and volumetric effects. In addition to these scenes, we further show results on the challenging *Parkour* scene, cf. Ch. 5.4. Fig. 8.5 shows still images of different effects created with our editor. Table 8.1 lists effect configuration details for the individual scenes and also shows how many additional space-time samples were synthesized per output frame.

Because we employ the well-known accumulation technique to emulate camera shutter effects, we must decide on a number of plenoptic samples N to be summed up in Eq.(8.2). Depending on output rendering resolution, GPU speed, and magnitude of scene motion, the number of samples is kept adaptive to ensure high-quality results without compromising performance. For the test scenes shown in the accompanying video, we have used $N = 25$ to obtain high-quality long-exposure effects. All space-time effects are visualized at frames rates exceeding 25 fps on an NVIDIA GeForce 260 GTX.

Output rendering quality obviously depends on the visual plausibility of the correspondence fields. We currently do not enforce strict consistency of correspondence fields around loops of tetrahedral edges of the navigation space tessellation described in Ch. 3.4. First steps in this direction have already been taken by Sellent et al. [133], however, extending the computation to loops involving more than three images could further stabilize correspondence estimation. Our automatic, pair-wise image-correspondence estimation algorithm described in Ch. 6 yields convincing and robust results overall, we explicitly allow for human interaction in addition to correct for re-

Scene	Cameras	Effects	Additional samples per frame
Firebreather	16	time& space ramps freeze-rotate & slow motion	0
Skateboarder	6	open flash & space blur freeze-rotate	5 + 25
Yuki	5	progressive motion distortion	121
Parkour	16	time/space blur & multi-exposure freeze-rotate & slow motion	25 + 20 + 3
Skateboarder	6	temporal flare freeze-rotate	0

Table 8.1: Effect configuration details for our test scenes, Fig. 8.5. Note that despite the high number of additional space-time samples needed for some effects, our STF/X editor still exceeds 25 fps.

maining spurious correspondences. Correspondence correction typically takes about one minute per video frame pair.

To ensure high-quality STF/X results, we observed that the angle between adjacent camcorders should not exceed ≈ 10 degrees, independent of scene content. For greater angular distances, missing scene information becomes apparently too large for contemporary interpolation algorithms. As a rule of thumb, we found that across space or time, scene correspondences should not be separated by more than approximately 20% of linear image size.

8.5 Conclusion

We have presented a space-time visual effects editor to create and interactively edit visual effects as a post-production process. Our approach allows creating space-time visual effects from sparse, unsynchronized multi-video footage of arbitrary, dynamic real-world scenes. We do not need special acquisition hardware or time-consuming setup procedures but record with consumer camcorders in arbitrary environments. A single multi-video recording suffices to generate arbitrary effects, providing high flexibility

(a) Open flash used as input for space blur.

(b) Space and time ramps.

(c) Scanline progressive motion distortion

(d) Multi-exposure.

(e) Temporal flare.

Figure 8.5: Examples of visual effects. The input data has been recorded with 5-16 Canon XHA1 camcorders. Each scene shows different visual effects.

and a considerable reduction in time and effort for STF/X production.

Future work will focus on adding high-level editing operations to our STF/X editor, like altering scene illumination and object appearance. Highly reliable dense correspondences will enable us to propagate image editing operators automatically to the entire set of multi-video images.

9

Flexible Stereoscopic 3D Content Creation of Real World Scenes

9.1 Introduction

The focus of the algorithms proposed for view synthesis so far was on visually plausible synthesis of in-between images. In this chapter we show that not only visually plausible, but also geometrically meaningful images are synthesized by the Virtual Video Camera by employing the system for the easy stereoscopic content creation.

Three major methodologies for creating stereoscopic material are currently employed: purely digital content creation for animation films and games, content filmed with specialized stereoscopic cameras, and stereo hallucination from monocular video. If the entire production process is digital in the sense that the shown scene exists as computer graphics models, the creation of a stereoscopic image pair is straightforward. Instead of rendering one image per video frame, a second image with a shifted virtual viewpoint is rendered. Since the recording environment including time is fully controllable, dynamic scenes do not pose an additional problem. The major drawback is, that the creation of naturalistic real world images is extremely complex and time consuming.

The enhancement of monocular recordings suffers from a similar problem. Although the recorded footage has the desired natural look, the creation of a proxy geometry or a scene model can be tedious. A depth map or proxy geometry for synthesizing a second viewpoint has to be created by hand modeling. Therefore the complexity of the scene model creation directly depends on the complexity of the recorded scene.

While directly recording with a stereoscopic camera eliminates the need to create an additional scene model, it requires the highly specialized and therefore expensive stereo-camera hardware. Leaving aside monetary constraints, the on set handling of the stereoscopic cameras poses a challenge. The view and baseline selection for example requires careful planning to give the viewer a pleasing stereoscopic experience. A parameter change in later production steps becomes very difficult.

We propose a purely image-based approach to solve the problem and thereby eliminate the requirement for an explicit scene model. By using the Virtual Video Camera [94], we show that it becomes possible to interpolate arbitrary stereoscopic views between the input camera viewpoints, extending conventional free-viewpoint video to the stereoscopic case. Since the Virtual Video Camera is capable to interpolate time and space , it combines the natural image impression from direct stereoscopic recording with the full viewpoint and time control of digitally created scenes. Further on, having full control over time and camera viewpoint, effect shots as a freeze-rotate are easily possible without the need for complex hardware setups. Even more, visual effects as presented in the last chapter can also be created for the stereoscopic case.

We start with an overview on related work in Ch. 9.2. We then discuss why the Virtual Video Camera system is able to create valid stereoscopic views by examining what is encoded in the correspondence fields in Ch. 9.3. An extended image formation and image-based rendering algorithm is described in Ch. 9.4. In Ch. 9.5 we show results that were created using only image-based techniques. For different scenarios geometrically plausible stereoscopic renderings are shown.

9.2 Related Work

Since its invention in 1838, stereoscopy has been widely used in photography and film making industry. It has recently received renewed attention, partly because sophisticated stereoscopic equipment became available for the consumer market. Although the basic principle of stereoscopic image acquisition seems quite simple, many pitfalls exist that make stereoscopic capture a tedious task. A good introduction to stereoscopic movie making is given by Mendiburu [109]. This book formalizes the concepts with clear and simple drawings and is considered as the reference in the movie-making community.

Similarly, Devernay and Beardsley give an in-depth review on the state of understanding in stereoscopic cinema [45]. In their article, they discuss perceptual factors, choice of camera geometry at capture time and post-production tools to manipulate 3D experience. A good introduction to current stereoscopic editing techniques can be found in Wilkes [165]. Typical stereoscopic editing task are image rectification, color balancing, disparity remapping and baseline editing. The latter one is especially interesting for our approach, since multi-view recordings often feature wide baselines and conversion to stereoscopic output material is not straight-forward. Rogmans et al. reviewed the available methods for novel view synthesis from stereoscopic data, and noticed that they essentially consist of two steps: disparity estimation and view synthesis [121]. Since disparity estimation is often error-prone, Devernay and Peon proposed a novel view synthesis for altering interocular distance with on-the-fly artifact detection and removal [47]. Disparity remapping also recently received considerable attention: Lang et al. proposed non-linear disparity mapping operators to alter perceived scene depth, necessary for content adaption to different viewing geometries [81]. Targeting the same application, Devernay and Duchêne proposed a disparity-remapping scheme that does not distort objects with respect to perceived depth [46]. To aid the stereographer in the first place with capture, Zilly et al. presented the Stereoscopic analyzer, a tool for online validation of stereoscopic capture, including image rectification, detection of window violations, and optimal interocular distance proposal [178]. Most similar to our proposed approach to stereoscopic content creation is the work of Guillemaut et al. [66]. They reconstruct a high-quality scene geometry from wide-baseline multi-view footage and use this geometry for stereoscopic view synthesis. While giving good results, they only demonstrated their approach on indoor scenes.

9.3 Correspondence Fields

To be able to create image-based stereoscopic images it is important to get an insight into the nature of the information contained in the correspondence fields $\mathbf{w}_{ij}$ from recorded source image I_i to a destination image I_j.

In order to get a clear understanding of the information encoded within $\mathbf{w}_{ij}$, we assume a setup consisting of two cameras. Figure 9.1 shows such a simplified setup where the cameras are restricted to a 1D movement along the horizontal axis. The

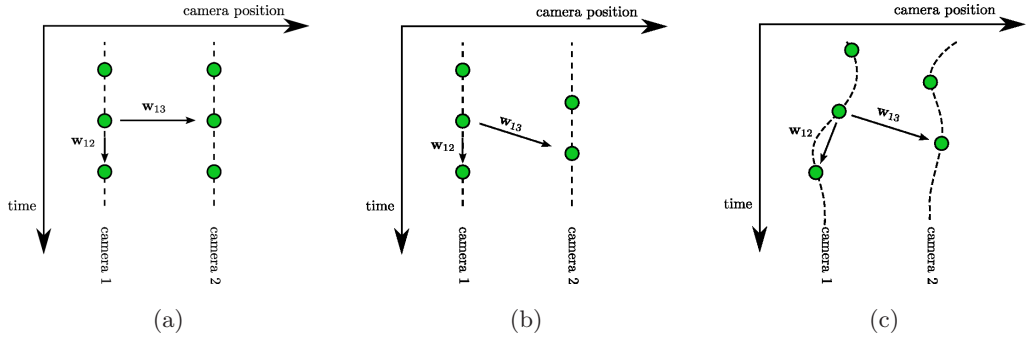


Figure 9.1: The layout of images shown as green dots in the space-time plane with different camera configurations: (a) static and synchronized cameras (b) static unsynchronized cameras (c) moving unsynchronized cameras.

analysis generalizes to 3D movement; for readability and visualization we stick with the 1D case. The vertical axis is the time, the dotted lines show the camera movement paths and each green dot corresponds to an image acquired at that specific time and place. In the following we investigate the information contained in the correspondence fields $\mathbf{w}_{12}$ and $\mathbf{w}_{13}$ with different recording constraints.

We start with the most restrictive camera setup with static cameras and synchronized camera shutters as shown in Fig. 9.1(a). The images are acquired on an axis aligned regular grid in the space-time plane. The correspondence field within the first camera $\mathbf{w}_{12}$ links two consecutive images of the video stream. Since the viewpoint does not change all image changes encoded within $\mathbf{w}_{12}$ represent motion of objects within the scene. In contrast the correspondence field $\mathbf{w}_{13}$ linking two adjacent cameras contains no object movement at all. Source and destination image have been acquired at the same point in time and therefore no object motion occurred. All changes along $\mathbf{w}_{13}$ can be explained by the motion parallax due to the changed viewpoint between cameras. If the source and destination image of $\mathbf{w}_{13}$ were rectified, the correspondence field would amount to a disparity map. While this is not true in the general case, $\mathbf{w}_{13}$ strictly encodes the scene depth information and no object movement.

When the recording constraints are relaxed to a setup where the camera shutters are no longer synchronized, the interpretation of $\mathbf{w}_{13}$ changes. While the intra camera correspondence field $\mathbf{w}_{12}$ still encodes only object motion, the inter camera correspondence field $\mathbf{w}_{13}$ is no longer horizontally aligned with the time axis. In the space-time plane this amounts to a sheering of the two camera paths as depicted in Fig. 9.1(b).

As a result of the unsynchronized shutters the objects in a dynamic scene can move between the acquisition times and the variation between source and destination image is no longer only defined by the view point change.

A similar change occurs when the cameras are allowed to change position. In the simplified one dimensional case the camera paths are no longer straight lines. As seen in Fig. 9.1(c) the direction from source to destination image for $\mathbf{w}_{12}$ is as well as $\mathbf{w}_{13}$ are no longer axis aligned. Both vector fields contain a mixture of scene depth and motion.

While we do not reconstruct an explicit depth or motion model of the scene at any point, the information is still encoded within the vector fields. This holds true for less simplified cases incorporating more cameras and view synthesis within $\mathcal{N}$.

Inherent to our model is however the restriction to linear motion. This applies on the one hand to the motion (Fig. 9.1(c)) of camera where $\mathbf{w}_{12}$ linearly connects source and destination images regardless of the actual camera motion. On the other hand it affects the object motion within the scene because a correspondence field defines a line along which each point can move on the image plane.

9.4 Stereoscopic Virtual View Synthesis

In this section, we briefly discuss how to use the Virtual Video Camera framework, Ch. 3, to synthesize stereoscopic views. The left view of the stereoscopic image pair is synthesized as introduced in Eq. (3.1) in Ch. 3.5.3. Again, disocclusions are detected on-the-fly by calculating local divergence in the correspondence fields. However, in contrast to the occlusion heuristics used in Ch. 3.5.3, we now determine a geometrically valid disparity map. When applying the forward warp in a vertex shader program, we also determine where a given vertex would be warped to in the right eye's view. By subtracting both values, we derive a per-vertex disparity. This quantity is interpolated in the fragment shader. Using this approximation of the disparity value, occlusions can now be resolved by a simple depth buffer comparison. If necessary, the exact disparity of a parallel camera setup can be computed by applying the identical post-warping re-projection matrices.

The view for the right eye $I_v^R(\varphi + \Delta, \theta, t)$ is synthesized similar to Eq. (3.1) by offsetting the camera position along the φ-direction. A common rule for stereoscopic

capture is that the maximal angle between the stereo camera axes in a converging setup should not exceed 1.5 degrees [109]. Otherwise, the eyes are forced to diverge to bring distant objects in alignment which usually causes discomfort. By nature of construction of $\mathcal{N}$, our approach renders converging stereo pairs and angles of convergence between 0.5 and 1.5 degrees give the most pleasing stereoscopic results.

9.4.1 Visual Effects and Disparity Effects

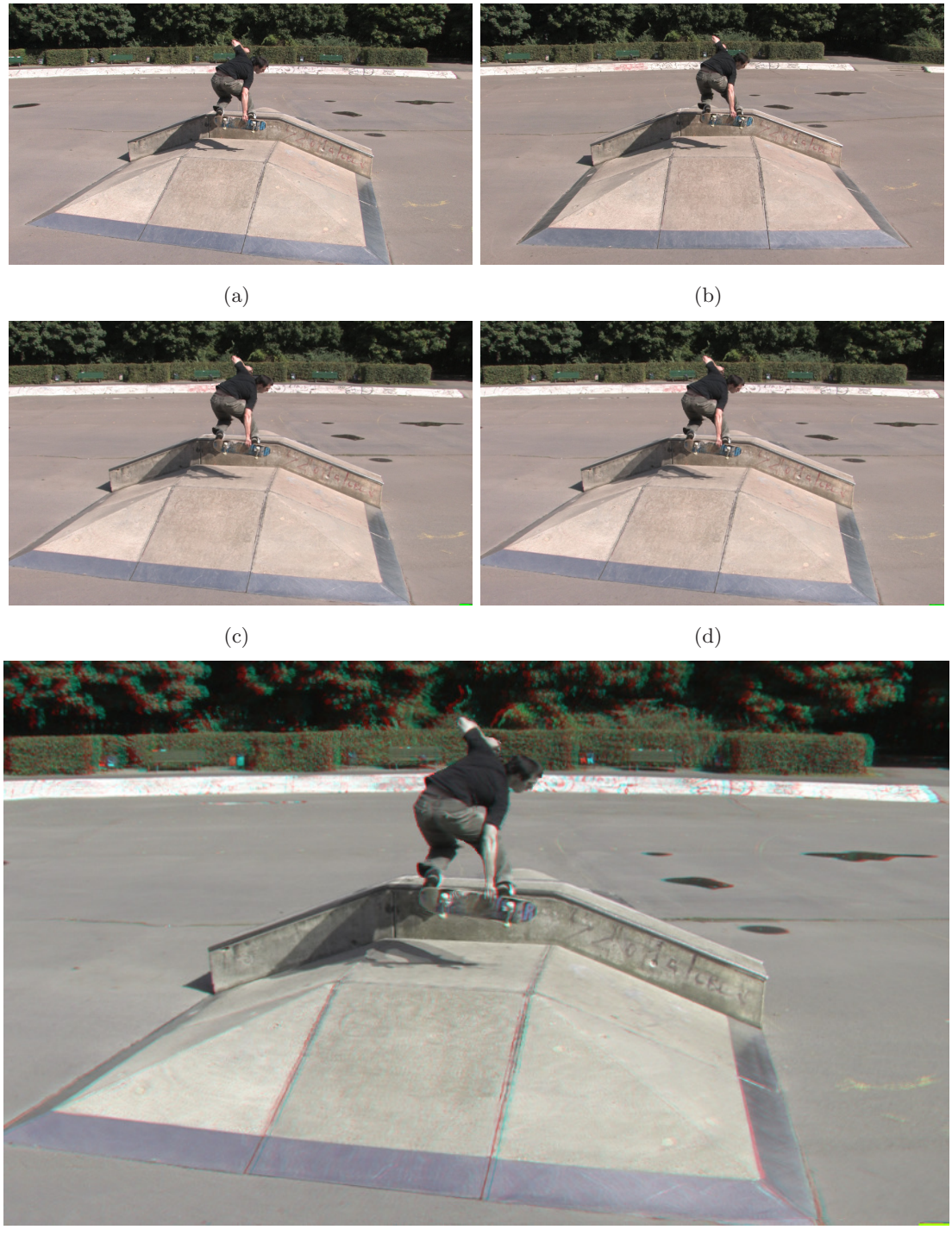
In the last chapter, we already showed how to integrate visual effects into the Virtual Video Camera. Since those effects can be created for any virtual viewpoint inside the navigation space, they can also be created for stereoscopic display.

Further, our system is able to produce disparity-based visual effects, since disparity maps for each stereo pair are calculated on-the-fly for occlusion handling. While image-based systems often lack this crucial ability, a vast amount of editing tasks becomes possible with on-the-fly creation of disparity maps. Typical editing tasks are insertion of 3D objects with correct depth ordering, synthetic depth-of-field rendering, atmospheric effects (attenuation/fog) and refocussing on the object of interest. In Ch. 9.5 we show a selection of these effects. Of course, using this image-based disparity, non-linear re-mapping as presented by Lang et al. [81] and Devernay and Duchêne [46] can be applied as well.

9.5 Results

All results in this section are presented in red (left) - cyan (right) anaglyph images. Again, we used the *Skateboarder* and *Parkour* sequences introduced in previous chapters to evaluate the performance of our approach. We further add the *Juggler* sequence, a multi-view dataset captured with hand-held cameras, and the *Breakdancer* sequence from Zitnick et al. [181] to our test set.

Figures 9.2(a) and (b) show the original input images for two adjacent cameras. The baseline in the setup is too wide and the image planes too different to make the original input views work as a stereo pair by themselves. In contrast, the final render results for the left and right stereo image in Figs. 9.2(c) and (d) are quite close. When watching the scene in motion, the stereo parallax effect of the actor in front of the

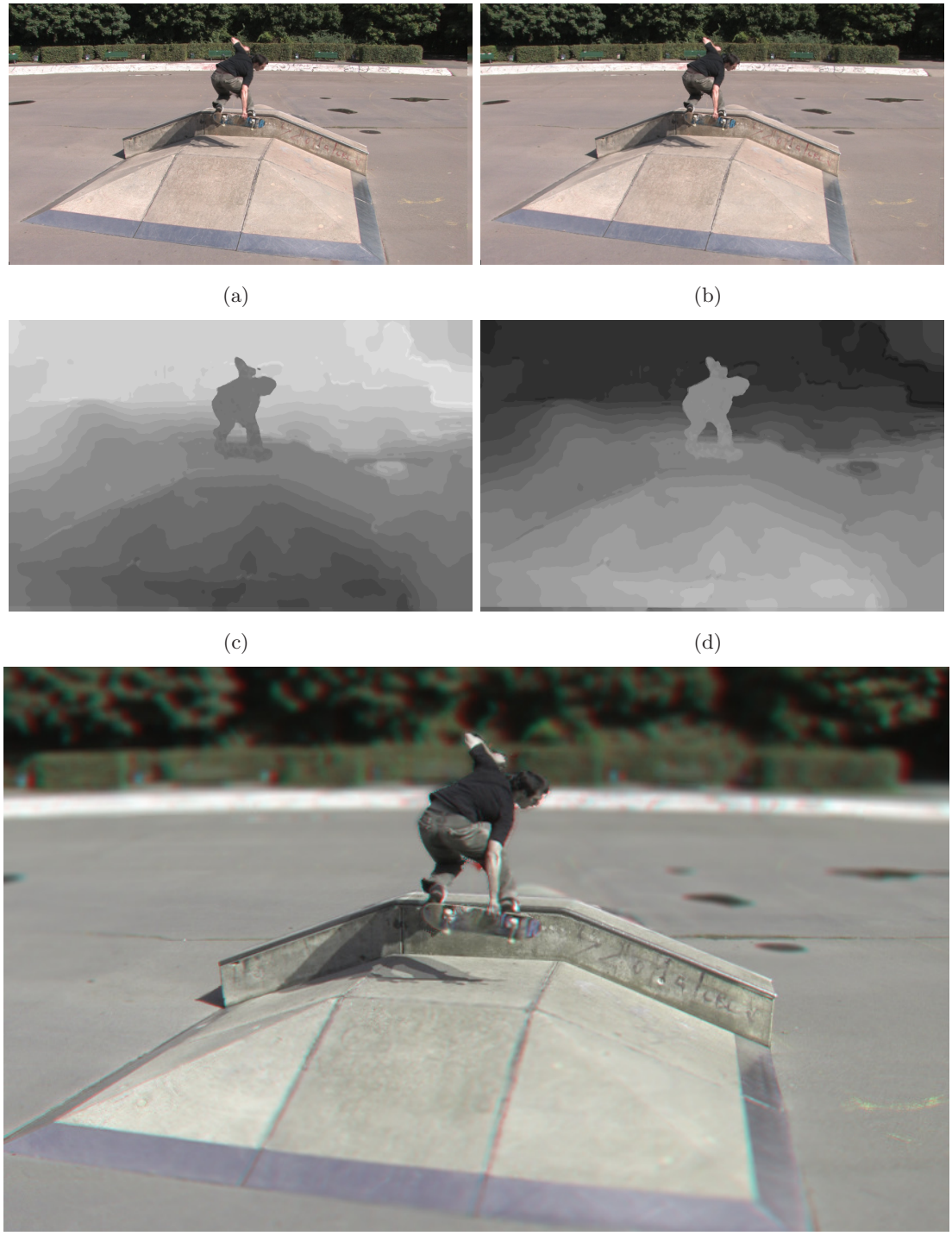


(a) (b)

(c) (d)

(e)

Figure 9.2: Multi-view Skater scene with one distinguished foreground object and large scene depth recorded with six cameras. (a),(b) Two original input views of adjacent cameras. (c),(d) Left and right view of stereo pair and (e) red-cyan anaglyph from virtual viewpoint. Please use red (left) - cyan (right) anaglyph glasses.



(a) (b)

(c) (d)

(e)

Figure 9.3: Depth-of-field effect. (a),(b) show the left and right image of the synthesized stereo pair, (c),(d) visualize the disparity map from left to right and vice versa and (e) shows a red-cyan anaglyph with a depth-of-field effect based on the estimated disparities. Please use red (left) - cyan (right) anaglyph glasses.

background trees clearly enhances the already existing motion parallax, to give a very good impression of scene depth.

The *Parkour* scene, Fig. 9.4(a), exhibits great detail in the background of the scene. The twigs of the trees are sometimes only of sub-pixel size, are visible at multiple depths and occlude each other. Although our rendering model does not explicitly cope with this fine geometry, the rendered results are convincing. When viewing the stereoscopic output video, the Parkour runner stands out clearly in front of the cluttered background, whereas he is more or less indistinguishable in a monocular sequence. It should also be apparent that it is very challenging to recover a 3D model of the scene. This applies equally to an automatic 3D reconstruction as well as to a manual modeling process. It can also be noted that the stereoscopic cues seem to convince the human eye of the plausibility of the scene. When watching a monocular rendering, rendering artifacts are spotted more easily. This is due to the fact that artifacts appearing in the left view not necessarily also appear in the (stereoscopic) correct position in the right view and are thus masked out.

In the *Juggler* scene, Fig. 9.4(b), we show that the stereoscopic content creation is also possible for moving cameras.

For comparative purposes, we again rendered a stereoscopic sequence with the *Breakdancer* input material from Zitnick et al. [181], Fig. 9.4(c).

We demonstrate a combination of two disparity-based effects on the *Skateboarder* scene. We refocus the stereoscopic image pair so that a virtual point of interest, i.e. a certain scene depth, has no disparity between the two eyes. A synthetic depth-of-field effect is further applied to steer the observer's attention towards these areas, Fig. 9.3(e). Stereoscopers may use similar effects to guide the observer's view and to strengthen or weaken the stereoscopic effect. We also integrated visual effects as described in Ch. 8 into our stereoscopic rendering. We rendered the *Skateboarder* scene with a time-freeze (Fig. 9.2(e)), time-blur (Fig. 9.4(e)) and flashtrail effect (Fig 9.4(d)) to show the versatility of our approach.

9.6 Conclusion

We presented an approach for stereoscopic free-viewpoint rendering that circumvents the need for explicit 3D reconstruction. It enables the flexible creation of stereoscopic

(a) Parkour scene.

(b) Juggler scene.

(c) Breakdancer scene from Zitnick et al. [181].

(d) Skateboarder scene with flashtrail effect.

(e) Skateboarder scene with timeblur effect.

Figure 9.4: Rendered anaglyph stereo images. With our approach, stereoscopic images can be rendered from wide-baseline multi-view recordings. Please use red (left) - cyan (right) anaglyph glasses.

content of complex natural scenes, where parameters as baseline, viewpoint and scene time can easily be modified in post production. In a single workflow, image alignment, free-viewpoint video and baseline editing can be performed.

Our approach can cope with asynchronously captured material and loosely calibrated camera setups greatly reducing the hardware requirements needed for stereoscopic 3D recording.

9. FLEXIBLE STEREOSCOPIC 3D CONTENT CREATION OF REAL WORLD SCENES

10

Discussion and Conclusions

In the following, we summarize our work and discuss contributions and limitations of the presented methods. We draw conclusions and present an outlook on future work.

10.1 Summary

First, we introduced multi-exposure images as a way to measure and analyze fast dynamic events. Our approach is based on deformable shape matching between the model shape in a conventional single exposure and the target shape in the multiply exposed image [92]. The shape models are built from edges detected in both images, the matching problem is solved as an assignment problem. A dense deformation field is then derived from the sparse matches by estimation of piecewise constant perspective transformations. We demonstrated that our approach yields plausible motion fields that can be used to synthesize in-between moments, e.g. for slow motion sequences. So far, we tested our method only on motion sequences that are fronto-parallel to the image plane and exhibit no occlusion. An extension to general motion sequences involving self-occlusions, however, is a non-trivial task.

Next, we evaluated the current state-of-the-art in optical flow research for use in our image-based space-time navigation system. We tested two top performers [143, 162] from the Middlebury evaluation benchmark [12], a long-range correspondence estimation method [136] and an edge-based method [140] on four scenes with increasing complexity. We assessed the quality of the resulting interpolation sequences in a perceptual user study. The study revealed that none of the tested algorithms could achieve

a perceptually plausible quality. Also, among the four algorithms tested, there was no statistically significant difference with respect to perceived quality.

We then combined the strengths of the evaluated approaches in an optical flow algorithm, tailored for image interpolation on complex natural scenes [88]. In particular, we base our algorithm on a long-range estimation technique avoiding a coarse-to-fine strategy to capture fine details and fast motion. We proposed a symmetric formulation and used SIFT descriptors in addition to color to more robustly resolve ambiguous matches. Our approach identifies occluded regions by examining symmetry of forward and backward motion vectors and corrects those motion vectors by means of geodesic inpainting [9]. We evaluated our proposed technique on the test scenes used for evaluation of the state-of-the-art. A psychophysical user study revealed that our approach yields higher interpolation quality than state-of-the-art on all scenes.

We examined blending-based multi-image interpolation more closely and derived a new formulation based on labeling [88]. By identifying a unique source for each pixel in the interpolated view, we are able to do without blending, and hence the low-pass filtering. Having found a source image for each pixel, we fill remaining holes in the interpolated view by spatio-temporal inpainting and reconstruct the interpolated image from gradient domain samples by solving the 3D Poisson equation [99]. We evaluated our approach on several test scenes and consistently maintained more high-frequency detail when compared to standard blending-based multi-image interpolation. The main drawback of our method lies in the computation time involved. This problem may be alleviated by transferring the labeling and reconstruction part of our algorithm to the GPU.

Finally, we put the correspondence estimation technique and the multi-image interpolation to work in the Virtual Video Camera, a system for purely image-based space-time navigation [94]. We propose an additional application scenario of the Virtual Video Camera: space-time visual effects design and stereoscopic view synthesis. First, we show how to shift visual effects design to post-production [90]. We show that with the Virtual Video Camera, we are able to synthesize various visual effects from a single recording. This is in contrast to the approach currently pursued by the film industry, where every individual visual effect is captured with a specific camera setup. Our approach thus greatly simplifies visual effects design and reduces the cost

for production. We then show that the Virtual Video Camera not only produces visually plausible but also geometrically valid images. To this end, we use the Virtual Video Camera system to produce stereoscopic content from various unsynchronized multi-view recordings. We further show that disparity maps between left and right view can be estimated on the fly during view synthesis.

10.2 Conclusions

In this thesis, we presented new methods for image-based, photo-realistic rendering of general, dynamic, real-world events. We focused on two particular aspects of image-based space-time navigation: methods for dense correspondence estimation, and high-frequency preserving image interpolation techniques. Our correspondence estimation technique is specifically tailored to the needs of image interpolation. We enforce symmetric correspondences in visible areas and use expressive SIFT descriptors to robustly match pixels over large displacements. Contrary to state-of-the-art optical flow algorithms, our correspondence estimation does not focus on numerically accurate correspondence fields but instead optimizes the interpolation error. Combined with label-based view synthesis, we are able to improve interpolation results compared to previously proposed space-time interpolation techniques. We demonstrated the applicability of our algorithms within the Virtual Video Camera and on several complex real-world scenes. Finally, we introduced two interesting applications, i.e. space-time visual effects design and stereoscopic content creation.

10.3 Future Work

Different lines of research can be conceived to further improve the quality and performance of the proposed methods. The multi-exposure technique could be integrated into an analysis-by-synthesis loop to further improve the quality of the resulting flow fields. An extension to multiple views might be beneficial for more complex motions including self occlusion; an extension to more general motions would also greatly increase the applicability of the proposed method.

For the label-based interpolation technique, research should be mostly focused on finding a more efficient way to solve the optimization and reconstruction problem.

A transfer of the algorithms to graphics hardware using CUDA or OpenCL appears promising.

Our dense correspondence estimation technique could be extended to simultaneously optimize for a per-pixel "photometric warp". This will help to adjust remaining color differences of matched pixels and to avoid color jumps when transitioning from one source image to the other in our label-based interpolation technique. It will also be interesting to combine dense correspondence estimation with scene geometry and scene motion estimation in a common framework.

For applications of the Virtual Video Camera, several research directions seem promising. First of all, a server-side rendering approach with streaming of the resulting interpolation is currently the only feasible way to handle the large amount of data needed by the system and might also increase the acceptance of the system. It would also alleviate the dependency on special graphics hardware currently employed by the system. Adding geometry to the system could help to avoid artifacts originating in the depth heuristics used during forward warping of the input images. It would further allow for view and time extrapolation. The flow fields and images encode all information needed to synthesize a novel view. It should also be possible to synthesize camera zooms by suitably combining the information contained in the images and correspondence fields. Finally, multi-view video editing, for example video matting or altering scene appearance, promises many interesting applications.

Bibliography

[1] Adelson, E. H. and Bergen, J. R. 1991. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*. MIT Press, 3–20.

[2] Adobe Inc. 2009. After Effects CS4.

[3] Aggarwal, J. and Nandhakumar, N. 1988. On the computation of motion from sequences of images: A review. In *Proceedings of the IEEE*. Vol. 76. 917–935.

[4] Akenine-Möller, T., Munkberg, J., and Hasselgren, J. 2007. Stochastic rasterization using time-continuous triangles. In *Proc. of Graphcis Hardware (GH'07)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 7–16.

[5] Alvarez, L., Deriche, R., Papadopoulo, T., and Sanchez, J. 2002. Symmetrical Dense Optical Flow Estimation with Occlusion Detection. In *Proc. of European Conference on Computer Vision (ECCV'02)*. Springer, 721–735.

[6] Atcheson, B., Heidrich, W., and Ihrke, I. 2009. An evaluation of optical flow algorithms for background oriented schlieren imaging. *Experiments in Fluids 46,* 3, 467–476.

[7] Atcheson, B., Ihrke, I., Heidrich, W., Tevs, A., Bradley, D., Magnor, M., and Seidel, H.-P. 2008. Time-resolved 3d capture of non-stationary gas flows. In *Proc. of ACM SIGGRAPH Asia 2008*. ACM, New York, NY, USA, 1–9.

[8] A.Wachowski and L.Wachowsky. 1999. Matrix. DVD. Warner Bros. Home Entertainment.

[9] Bai, X. and Sapiro, G. 2009. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *IJCV 82*, 113–132. 10.1007/s11263-008-0191-z.

[10] Bai, X., Wang, J., Simons, D., and Sapiro, G. 2009. Video SnapCut: robust video object cutout using localized classifiers. *ACM Trans. on Graphics 28,* 3, 70:1–70:11.

[11] Baker, S., Roth, S., Scharstein, D., Black, M. J., Lewis, J., and Szeliski, R. 2007. A database and evaluation methodology for optical flow. In *IEEE International Conference on Computer Vision (ICCV'07)*. IEEE Computer Society, Los Alamitos, CA, USA, 1–8.

[12] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., and Szeliski, R. 2010. Middlebury Optical Flow Evaluation Database. http://vision.middlebury.edu/flow.

[13] Ballan, L., Brostow, G. J., Puwein, J., and Pollefeys, M. 2010. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM Trans. on Graphics 29,* 3 (July), 87:1–87:11.

[14] Barron, J., Fleet, D., and Beauchemin, S. 1994. Performance of optical flow techniques. *IJCV 12,* 1, 43–77.

[15] Battiti, R., Amaldi, E., and Koch, C. 1991. Computing optical flow across multiple scales: An adaptive coarse-to-fine strategy. *IJCV 6,* 2 (June), 133–145.

[16] Beauchemin, S. S. and Barron, J. L. 1995. The computation of optical flow. *ACM Comput. Surv. 27,* 3, 433–466.

[17] Beier, T. and Neely, S. 1992. Feature-based image metamorphosis. *Computer Graphics (Proc. of SIGGRAPH 93) 26,* 2, 35–42.

[18] Belongie, S., Malik, J., and Puzicha, J. 2001. Matching shapes. In *IEEE International Conference on Computer Vision (ICCV'01)*. Vol. 1. IEEE Computer Society, 454–463.

[19] Bergen, J. R., Anandan, P., Hanna, K. J., and Hingorani, R. 1992. Hierarchical model-based motion estimation. In *Proc. of European Conference on Computer Vision (ECCV'92)*. Springer-Verlag, London, UK, 237–252.

[20] Berkels, B., Kondermann, C., Garbe, C., and Rumpf, M. 2009. Reconstructing Optical Flow Fields by Motion Inpainting. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*. Lecture Notes in Computer Science, vol. 5681. Springer, 388–400.

[21] Bertsekas, D. P. 1988. The auction algorithm: a distributed relaxation method for the assignment problem. *Ann. Oper. Res. 14,* 1-4, 105–123.

[22] Besl, P. J. and McKay, N. D. 1992. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 14,* 2, 239–256.

[23] Bhat, P., Zitnick, C. L., Snavely, N., Agarwala, A., Agrawala, M., Curless, B., Cohen, M., and Kang, S. B. 2007. Using photographs to enhance videos of a static scene. In *Proc. of Eurographics Rendering Workshop*, J. Kautz and S. Pattanaik, Eds. Eurographics, 327–338.

[24] Black, M. and Anandan, P. 1996. The robust estimation of multiple motions: Parametric and piecewise-smooth flow-fields. *Computer Vision and Image Understanding 63,* 1 (January), 75–104.

[25] Bookstein, F. L. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell. 11,* 6, 567–585.

[26] Bousseau, A., Neyret, F., Thollot, J., and Salesin, D. 2007. Video watercolorization using bidirectional texture advection. *ACM Trans. on Graphics 26,* 3, 104.

[27] Box, G. E., Hunter, J. S., and Hunter, W. G. 2005. *Statistics for experimenters: Design, Innovation, and Discovery.* Wiley-Interscience; 2 edition.

[28] Boykov, Y., Veksler, O., and Zabih, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell. 23,* 11, 1222–1239.

[29] Brostow, G. J. and Essa, I. 2001. Image-based motion blur for stop motion animation. In *Proc. of ACM SIGGRAPH 2001.* ACM, New York, NY, USA, 561–566.

[30] Brox, T., Bregler, C., and Malik, J. 2009. Large Displacement Optical Flow. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'09)*. to appear.

[31] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. 2004. High accuracy optical flow estimation based on a theory for warping. In *Proc. of European Conference on Computer Vision (ECCV'04)*. Vol. 4. 25–36.

[32] Bruhn, A., Weickert, J., and Schnörr, C. 2005. Lucas/kanade meets horn/schunck: combining local and global optic flow methods. *IJCV 61,* 3, 211–231.

[33] Buehler, C., Bosse, M., McMillan, L., Gortler, S., and Cohen, M. 2001. Unstructured Lumigraph Rendering. In *Proc. of ACM SIGGRAPH 2001*. ACM Press/ACM SIGGRAPH, New York, 425–432.

[34] Cammarano, M. and Jensen, H. W. 2002. Time dependent photon mapping. In *Proc. of Eurographics Rendering Workshop*. 135–144.

[35] Carranza, J., Theobalt, C., Magnor, M., and Seidel, H. P. 2003. Free-Viewpoint Video of Human Actors. *ACM Trans. on Graphics 22,* 3, 569–577.

[36] Catmull, E. and Rom, R. 1974. A class of local interpolating splines. In *Computer Aided Geometric Design*, R. Barnhill and R. Riesenfeld, Eds. Academic Press, 317–326.

[37] Chambolle, A. 2004. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis. 20,* 1-2, 89–97.

[38] Chen, S. E. and Williams, L. 1993. View interpolation for image synthesis. In *Proc. of ACM SIGGRAPH 1993*. ACM Press/ACM SIGGRAPH, New York, 279–288.

[39] Chuang, Y.-Y., Agarwala, A., Curless, B., Salesin, D. H., and Szeliski, R. 2002. Video matting of complex scenes. *ACM Trans. on Graphics 21,* 3 (July), 243–248. Special Issue of the SIGGRAPH 2002 Proceedings.

[40] Chui, H. and Rangarajan, A. 2000. A new algorithm for non-rigid point matching. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*. IEEE Computer Society, 44–51.

[41] Cootes, T. and Taylor, C. J. 1992. Active shape models - 'smart snakes. In *Proc. of British Machine Vision Conference*. Springer-Verlag, 266–275.

[42] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. 2001. *Introduction to Algorithms, Second edition*. The MIT Press, Cambrigde, Massachusetts.

[43] de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.-P., and Thrun, S. 2008. Performance Capture from Sparse Multi-View Video. *ACM Trans. on Graphics 27,* 3, 98:1–98:10.

[44] Debevec, P., Borshukov, G., and Yu, Y. 1998. Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping. In *Proc. of Eurographics Rendering Workshop*. Eurographics Association, 105–116.

[45] Devernay, F. and Beardsley, P. 2010. *Image and Geometry Processing for 3-D Cinematography*. Vol. 5. Springer-Verlag, Chapter Stereoscopic cinema, 11–51.

[46] Devernay, F. and Duchêne, S. 2010. New view synthesis for stereo cinema by hybrid disparity remapping. In *Proc. of IEEE International Conference on Image Processing (ICIP'10)*. IEEE Computer Society, Hong Kong, 5–8.

[47] Devernay, F. and Peon, A. R. 2010. Novel view synthesis for stereoscopic cinema: Detecting and removing artifacts. In *Proc. of ACM Workshop on 3D Video Processing (3DVP'10)*. ACM Press, Firenze, 25–30.

[48] Digital Air Inc. 2009. Digital Air Techniques. http://www.digitalair.com/techniques/index.html.

[49] Egan, K., Tseng, Y.-T., Holzschuch, N., Durand, F., and Ramamoorthi, R. 2009. Frequency analysis and sheared reconstruction for rendering motion blur. *ACM Trans. on Graphics 28,* 3, 93:1–93:13.

[50] Einarsson, P., Chabert, C.-F., Jones, A., Lamond, B., Ma, W.-C., Sylwan, S., Hawkins, T., and Debevec, P. 2006. Relighting Human Locomotion

with Flowed Reflectance Fields. In *Proc. of Eurographics Rendering Workshop*. Eurographics Association, 183–194.

[51] Eisemann, E. and Durand, F. 2004. Flash photography enhancement via intrinsic relighting. *ACM Trans. on Graphics 23,* 3, 673–678.

[52] Eisemann, M., Decker, B. D., Magnor, M., Bekaert, P., de Aguiar, E., Ahmed, N., Theobalt, C., and Sellent, A. 2008. Floating Textures. *Computer Graphics Forum 27,* 2 (4), 409–418.

[53] Eisemann, M., Wolf, J., and Magnor, M. 2009. Spectral Video Matting. In *Proc. of Vision, Modeling and Visualization (VMV 2009)*. Braunschweig, Germany, 121–126.

[54] Enkelmann, W. 1988. Investigation of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics, and Image Processing 43,* 2 (August), 150–177.

[55] Felzenszwalb, P. F. and Huttenlocher, D. P. 2004. Efficient graph-based image segmentation. *IJCV 59,* 2, 167–181.

[56] Fujii, T. and Tanimoto, M. 2002. Free viewpoint TV system based on ray-space representation. In *Proc. of SPIE*. Vol. 4864. SPIE, 175.

[57] Germann, M., Hornung, A., Keiser, R., Ziegler, R., Würmlin, S., and Gross, M. 2010. Articulated billboards for video-based rendering. *Computer Graphics Forum 29,* 2, 585–594.

[58] Glocker, B., Paragios, N., Komodakis, N., Tziritas, G., and Navab, N. 2008. Optical flow estimation with uncertainties through dynamic mrfs. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'08)*. 1–8.

[59] Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S. 2007. Multi-view stereo for community photo collections. In *IEEE International Conference on Computer Vision (ICCV'07)*. 1–8.

[60] Gold, S., Rangarajan, A., Lu, C., Pappu, S., and Mjolsness, E. 1998. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern Recognition 31,* 8, 1019–1031.

[61] Goldlücke, B., Ihrke, I., Linz, C., and Magnor, M. 2007. Weighted Minimal Hypersurface Reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell. 29,* 7 (July), 1194–1208.

[62] Goldlücke, B. and Magnor, M. 2004. Space-time Isosurface Evolution for Temporally Coherent 3D Reconstruction. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*. Vol. I. 350–355.

[63] Goldlücke, B., Magnor, M., and Wilburn, B. 2002. Hardware-accelerated dynamic light field rendering. In *Proc. of Vision, Modeling and Visualization (VMV 2002)*. AKA, Heidelberg, 455–462.

[64] Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. 1996. The lumigraph. In *Proc. of ACM SIGGRAPH 1996*. ACM, New York, NY, USA, 43–54.

[65] Grundland, M., Vohra, R., Williams, G. P., and Dodgson, N. A. 2006. Cross Dissolve without Cross Fade: Preserving Contrast, Color and Salience in Image Compositing. *Computer Graphics Forum*, 577–586.

[66] Guillemaut, J.-Y., Sarim, M., and Hilton, A. 2010. Stereoscopic content production of complex dynamic scenes using a wide-baseline monoscopic camera set-up. In *Proc. of IEEE International Conference on Image Processing (ICIP'10)*. IEEE Computer Society, 9–12.

[67] Haeberli, P. and Akeley, K. 1990. The accumulation buffer: hardware support for high-quality rendering. In *Computer Graphics (Proceedings of SIGGRAPH 90)*. ACM, New York, NY, USA, 309–318.

[68] Hartley, R. and Zisserman, H. 2000. *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[69] Hasler, N., Rosenhahn, B., Thormählen, T., Wand, M., Gall, J., and Seidel, H.-P. 2009. Markerless Motion Capture with Unsynchronized Moving Cameras. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE Computer Society, Washington, 224–231.

[70] Hays, J. and Essa, I. 2004. Image and video based painterly animation. In *Proc. of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR'04)*. ACM, New York, NY, USA, 113–120.

[71] Hilsmann, A. and Eisert, P. 2008. Optical flow based tracking and retexturing of garments. In *Proc. of IEEE International Conference on Image Processing (ICIP'08)*. San Diego, California, USA.

[72] Hilsmann, A. and Eisert, P. 2009. Realistic cloth augmentation in single view video. In *Proc. of Vision, Modeling and Visualization (VMV 2009)*. Braunschweig, Germany.

[73] Horn, B. K. P. and Schunck, B. G. 1981. Determining optical flow. *Artificial Intelligence 17*, 185–203.

[74] Hornung, A. and Kobbelt, L. 2009. Interactive pixel-accurate free viewpoint rendering from images with silhouette aware sampling. *Computer Graphics Forum 28,* 8, 2090 – 2103.

[75] Hubel, D. H. 1995. *Eye, Brain and Vision, 2nd ed.* W.H. Freeman.

[76] Ince, S. and Konrad, J. 2008. Occlusion-Aware Optical Flow Estimation. *IEEE Trans. Image Processing 17,* 8, 1443–1451.

[77] Kang, S., Uttendaele, M., Winder, S., and Szeliski, R. 2003. High Dynamic Range Video. *ACM Trans. on Graphics 22,* 3, 319–325.

[78] Kim, B. and Essa, I. 2005. Video-based nonphotorealistic and expressive illustration of motion. In *Proc. of Computer Graphics International 2005*. IEEE Computer Society, Washington, DC, USA, 32–35.

[79] Klose, F., Linz, C., Lipski, C., and Magnor, M. 2010. Flexible stereoscopic 3d content creation of real world scenes. Tech. Rep. 2010-11-14, Computer Graphics Lab, TU Braunschweig, http://www.digibib.tu-bs.de/?docid=00036634. November.

[80] Klose, F., Lipski, C., and Magnor, M. 2010. Reconstructing Shape and Motion from Asynchronous Cameras. In *Proc. of Vision, Modeling and Visualization (VMV 2010)*. Eurographics, Eurographics Association, Siegen, Germany, 171–177.

[81] Lang, M., Hornung, A., Wang, O., Poulakos, S., Smolic, A., and Gross, M. 2010. Nonlinear disparity mapping for stereoscopic 3d. *ACM Trans. on Graphics 29,* 3, 75:1–75:10.

[82] Lee, S., Wolberg, G., and Shin, S. 1998. Polymorph: morphing among multiple images. *IEEE Computer Graphics and Applications*, 58–71.

[83] Lempitsky, V., Roth, S., and Rother, C. 2008. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'08)*. 1–8.

[84] Lempitsky, V., Rother, C., and Blake, A. 2007. Logcut - efficient graph cut optimization for markov random fields. In *IEEE International Conference on Computer Vision (ICCV'07)*. 1 –8.

[85] Leordeanu, M. and Hebert, M. 2005. A spectral technique for correspondence problems using pairwise constraints. In *IEEE International Conference on Computer Vision (ICCV'05)*. IEEE Computer Society, Washington, DC, USA, 1482–1489.

[86] Levoy, M. and Hanrahan, P. 1996. Light Field Rendering. In *Proc. of ACM SIGGRAPH 1996*. ACM Press/ACM SIGGRAPH, New York, 31–42.

[87] Li, Y. and Huttenlocher, D. P. 2008. Learning for optical flow using stochastic optimization. In *Proc. of European Conference on Computer Vision (ECCV'08)*. Springer-Verlag, Berlin, Heidelberg, 379–391.

[88] Linz, C., Lipski, C., and Magnor, M. 2010a. Multi-image Interpolation based on Graph-Cuts and Symmetric Optic Flow. In *Proc. of Vision, Modeling and Visualization (VMV 2010)*, C. R.-S. Reinhard Koch, Andreas Kolb, Ed. Eurographics, Eurographics Association, Siegen, Germany, 115–122.

[89] Linz, C., Lipski, C., and Magnor, M. 2010b. Multi-Image Interpolation based on Graph-cuts and Symmetric Optical Flow. Poster at SIGGRAPH 2010.

[90] Linz, C., Lipski, C., Rogge, L., Theobalt, C., and Magnor, M. 2010. Space-Time visual effects as a Post-Production process. In *Proc. of ACM Workshop on 3D Video Processing (3DVP'10)*. Firenze, Italy, 1–6.

[91] Linz, C. and Magnor, M. 2010. Dense correspondence estimation for image interpolation. Tech. Rep. 2010-11-13, Computer Graphics Lab, TU Braunschweig, http://www.digibib.tu-bs.de/?docid=00036631. November.

[92] Linz, C., Stich, T., and Magnor, M. 2008. High-speed Motion Analysis with Multi-exposure Images. In *Proc. of Vision, Modeling, and Visualization (VMV 2008)*, O. Deussen, D. Saupe, and D. Keim, Eds. Konstanz, Germany, 273–281.

[93] Lipski, C., Bose, D., Eisemann, M., Berger, K., and Magnor, M. 2010. Sparse Bundle Adjustment Speedup Strategies. In *WSCG Short Papers Post-Conference Proceedings*, V. Skala, Ed. WSCG.

[94] Lipski, C., Linz, C., Berger, K., Sellent, A., and Magnor, M. 2010. Virtual video camera: Image-based viewpoint navigation through space and time. *Computer Graphics Forum 29,* 8 (Dezember), 2555–2568.

[95] Lipski, C., Linz, C., Neumann, T., and Magnor, M. 2010. High Resolution Image Correspondences For Video Post-Production. In *Proc. of European Conference on Visual Media Production (CVMP'10)*. London, 33–39.

[96] Liu, C., Yuen, J., Torralba, A., Sivic, J., and Freeman, W. T. 2008. Sift flow: Dense correspondence across different scenes. In *Proc. of European Conference on Computer Vision (ECCV'08)*. Springer-Verlag, Berlin, Heidelberg, 28–42.

[97] Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *IJCV 60,* 2, 91–110.

[98] Lucas, B. and Kanade, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proc. of International Joint Conference on Artificial Intelligence*. Vancouver, Canada, 674–679.

[99] Mahajan, D., Huang, F., Matusik, W., Ramamoorthi, R., and Belhumeur, P. 2009. Moving Gradients: A Path-Based Method for Plausible Image Interpolation. *ACM Trans. on Graphics 28,* 3, 42:1–42:11.

[100] Manning, R. and Dyer, C. 1999. Interpolating View and Scene Motion by Dynamic View Morphing. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*. IEEE Computer Society, Washington, 388–394.

[101] Mark, W., McMillan, L., and Bishop, G. 1997. Post-Rendering 3D Warping. In *Proc. of Symposium on Interactive 3D Graphics (I3D'97)*. 7–16.

[102] Marr, D. and Hildreth, E. 1980. Theory of Edge Detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences 207,* 1167, 187–217.

[103] Matsuyama, T., Wu, X., Takai, T., and Nobuhara, S. 2004. Real-time 3D shape reconstruction, dynamic 3D mesh deformation, and high fidelity visualization for 3D video. *Computer Vision and Image Understanding 96,* 3, 393–434.

[104] Matusik, W., Buehler, C., Raskar, R., Gortler, S., and McMillan, L. 2000. Image-Based Visual Hulls. In *Proc. of ACM SIGGRAPH 2000*. ACM Press/ACM SIGGRAPH, New York, 369–374.

[105] Matusik, W. and Pfister, H. 2004. 3D TV: A Scalable System for Real-Time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes. *ACM Trans. on Graphics 23,* 3, 814–824.

[106] Max, N. L. and Lerner, D. M. 1985. A two-and-a-half-d motion-blur algorithm. In *Computer Graphics (Proceedings of SIGGRAPH 85)*. Vol. 19. ACM, New York, NY, USA, 85–93.

[107] McCane, B., Novins, K., Crannitch, D., and Galvin, B. 2001. On benchmarking optical flow. *Computer Vision and Image Understanding 84*, 126–143.

[108] McMillan, L. and Bishop, G. 1995. Plenoptic Modeling. In *Proc. of ACM SIGGRAPH 1995*. ACM Press/ACM SIGGRAPH, New York, 39–46.

[109] Mendiburu, B. 2009. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press.

[110] Meyer, B., Stich, T., Magnor, M., and Pollefeys, M. 2008. Subframe Temporal Alignment of Non-Stationary Cameras. In *Proc. of British Machine Vision Conference*. 103–112.

[111] Mitiche, A. and Bouthemy, P. 1996. Computation and analysis of image motion: A synopsis of current problems of methods. *IJCV 19,* 1, 29–55.

[112] Mori, G. and Malik, J. 2002. Estimating human body configurations using shape context matching. In *Proc. of European Conference on Computer Vision (ECCV'02)*. Vol. 3. 666–680.

[113] Munkres, J. 1957. Algorithms for the Assignment and Transportation Problems. *Journal of the Society of Industrial and Applied Mathematics 5,* 1, 32–38.

[114] Naemura, T., Tago, J., and Harashima, H. 2002. Real-Time Video-Based Modeling and Rendering of 3D Scenes. *IEEE Computer Graphics and Applications 22,* 2, 66–73.

[115] Nagel, H. H. and Enkelmann, W. 1986. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Anal. Mach. Intell. 8,* 5, 565–593.

[116] O'Neil, E. J., O'Neil, P. E., and Weikum, G. 1993. The lru-k page replacement algorithm for database disk buffering. *SIGMOD Rec. 22,* 2, 297–306.

[117] Otte, M. and Nagel, H.-H. 1994. Optical flow estimation: advances and comparisons. In *Proc. of European Conference on Computer Vision (ECCV'94)*. 51–60.

[118] Peers, P., Tamura, N., Matusik, W., and Debevec, P. 2007. Post-production Facial Performance Relighting using Reflectance Transfer. *ACM Trans. on Graphics 26,* 3, 52–62.

[119] Perona, P. and Malik, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell. 12,* 7, 629–639.

[120] Potmesil, M. and Chakravarty, I. 1983. Modeling motion blur in computer-generated images. In *Computer Graphics (Proceedings of SIGGRAPH 83)*. Vol. 17. ACM, New York, NY, USA, 389–399.

[121] Rogmans, S., Lu, J., Bekaert, P., and Lafruit, G. 2009. Real-time stereo-based view synthesis algorithms: A unified framework and evaluation on commodity

gpus. *Signal Processing: Image Communication 24,* 1-2, 49 – 64. Special issue on advances in three-dimensional television and video.

[122] Roth, S. and Black, M. J. 2007. On the Spatial Statistics of Optical Flow. *IJCV 74,* 1, 33–50.

[123] Ruzon, M. A. and Tomasi, C. 1999. Color edge detection with the compass operator. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*. IEEE Computer Society, 2160–2166.

[124] Sand, P. and Teller, S. 2004. Video matching. *ACM Trans. on Graphics 23,* 3, 592–599.

[125] Sand, P. and Teller, S. 2008. Particle Video: Long-Range Motion Estimation Using Point Trajectories. *IJCV 80,* 1 (October), 72–91.

[126] Schaefer, S., McPhail, T., and Warren, J. 2006. Image deformation using moving least squares. *ACM Trans. on Graphics 25,* 3, 533–540.

[127] Schirmacher, H., Heidrich, W., and peter Seidel, H. 2000. High-quality interactive lumigraph rendering through warping. In *Proc. of Graphics Interface (GI'00)*. CHCCS, 87–94.

[128] Scholz, V. and Magnor, M. 2004. Cloth motion from optical flow. In *Proc. of Vision, Modeling, and Visualization (VMV 2004)*. 117–123.

[129] Schwartz, C. and Klein, R. 2009. Improving initial estimations for structure from motion methods. In *Proc. of the Central European Seminar on Computer Graphics (CESCG).*

[130] Seitz, S. M. and Dyer, C. R. 1996. View Morphing. In *Proc. of ACM SIGGRAPH 1996*. ACM Press/ACM SIGGRAPH, New York, 21–30.

[131] Sellent, A., Eisemann, M., Goldlücke, B., Pock, T., Cremers, D., and Magnor, M. 2009. Variational Optical FLow from Alternate Exposure Images. In *Proc. of Vision, Modeling and Visualization (VMV 2009).*

[132] Sellent, A., Eisemann, M., and Magnor, M. 2009. Motion Field and Occlusion Time Estimation via Alternate Exposure Flow. In *Proc. of IEEE International Conference on Computational Photography (ICCP'09)*. IEEE.

[133] Sellent, A., Linz, C., and Magnor, M. 2010. Consistent Optical Flow for Stereo Video. In *Proc. of IEEE International Conference on Image Processing (ICIP'10)*. to appear.

[134] Snavely, N., Seitz, S. M., and Szeliski, R. 2006. Photo tourism: Exploring photo collections in 3d. *ACM Trans. on Graphics 25,* 3, 835–846.

[135] Starck, J. and Hilton, A. 2007. Surface Capture for Performance Based Animation. *IEEE Computer Graphics and Applications 27,* 3, 21–31.

[136] Steinbrücker, F., Pock, T., and Cremers, D. 2009. Large Displacement Optical Flow Computation without Warping. In *IEEE International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan.

[137] Steinbrücker, F., T.Pock, and D.Cremers. 2009. Advanced Data Terms for Variational Optic Flow Estimation. In *Proc. of Vision, Modeling and Visualization (VMV 2009)*. 155–162.

[138] Stich, T. 2009. Space-Time Interpolation Techniques. Ph.D. thesis, Institut für Computergraphik, Carl-Friedrich-GaußFakultät, Technische Universität Carola-Wilhelmina zu Braunschweig.

[139] Stich, T., Linz, C., Albuquerque, G., and Magnor, M. 2008. View and Time Interpolation in Image Space. *Computer Graphics Forum 27,* 7, 1781–1787.

[140] Stich, T., Linz, C., Wallraven, C., Cunningham, D., and Magnor, M. 2010. Perception-motivated Interpolation of Image Sequences. *ACM Transactions on Applied Perception (TAP)*, to appear.

[141] Stich, T., Linz, C., Wallraven, C., Cunningham, D., and Magnorw, M. 2008. Perception-motivated Interpolation of Image Sequences. In *Proc. of ACM Symposium on Applied Perception in Graphics and Visualization (APGV)*. ACM, Los Angeles, USA.

[142] Stiller, C. and Konrad, J. 1999. Estimating motion in image sequences: A tutorial on modeling and computation of 2d motion. *IEEE Signal Processing Magazine 16,* 4, 70–91.

[143] Sun, D., Roth, S., and Black, M. J. 2010. Secrets of optical flow estimation and their principles. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'10)*. IEEE Computer Society, 2432–2439.

[144] Sun, D., Roth, S., Lewis, J. P., and Black, M. J. 2008. Learning optical flow. In *Proc. of European Conference on Computer Vision (ECCV'08)*. Springer-Verlag, Berlin, Heidelberg, 83–97.

[145] Telea, A. 2004. An image inpainting technique based on the fast marching method. *Journal of Graphical Tools Vol.9,* No.1, 25–36.

[146] Telleen, J., Sullivan, A., Yee, J., Wang, O., Gunawardane, P., Collins, I., and Davis, J. 2007. Synthetic Shutter Speed Imaging. *Computer Graphics Forum 26,* 3, 591–598.

[147] Thayananthan, A., Stenger, B., Torr, P., and Cipolla, R. 2003. Shape context and chamfer matching in cluttered scenes. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*. IEEE Computer Society, 127–133.

[148] Theobalt, C., Albrecht, I., Haber, J., Magnor, M., and Seidel, H.-P. 2004. Pitching a Baseball – Tracking High-Speed Motion with Multi-Exposure Images. *ACM Trans. on Graphics 23,* 3, 540–547.

[149] Trobin, W., Pock, T., Cremers, D., and Bischof, H. 2008a. Continuous Energy Minimization Via Repeated Binary Fusion . In *Proc. of European Conference on Computer Vision (ECCV'08)*. Vol. 5305. 677–690.

[150] Trobin, W., Pock, T., Cremers, D., and Bischof, H. 2008b. An unbiased second-order prior for high-accuracy motion estimation. In *Proc. of DAGM Symposium on Pattern Recognition*. Springer-Verlag, Berlin, Heidelberg, 396–405.

[151] Vaudrey, T., Rabe, C., Klette, R., and Milburn, J. 2008. Differences Between Stereo and Motion Behaviour on Synthetic and Real-World Stereo Sequences. In *Proc. of International Conference on Image and Vision Computing New Zealand (IVCNZ'08)*.

[152] Vedula, S., Baker, S., and Kanade, T. 2005. Image Based Spatio-Temporal Modeling and View Interpolation of Dynamic Events. *ACM Trans. on Graphics 24,* 2, 240–261.

[153] Wang, H., Sun, M., and Yang, R. 2007. Space-Time Light Field Rendering. *IEEE Trans. Visualization and Computer Graphics 13,* 4, 697–710.

[154] Wang, H. and Yang, R. 2005. Towards space: time light field rendering. In *Proc. of Symposium on Interactive 3D Graphics (I3D'05)*. ACM Press, New York, 125–132.

[155] Wang, J., Xu, Y., Shum, H.-Y., and Cohen, M. F. 2004. Video tooning. *ACM Trans. on Graphics 23,* 3, 574–583.

[156] Waschbüsch, M., Würmlin, S., Cotting, D., Sadlo, F., and Gross, M. 2005. Scalable 3d video of dynamic scenes. *The Visual Computer 21,* 8, 629–638.

[157] Wedel, A., Cremers, D., Pock, T., and Bischof, H. 2009. Structure- and motion-adaptive regularization for high accuracy optic flow. In *IEEE International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan.

[158] Wedel, A., Pock, T., Braun, J., Franke, U., and Cremers, D. 2008. Duality tv-l1 flow with fundamental matrix prior. In *Proc. of International Conference on Image and Vision Computing New Zealand (IVCNZ'08)*. 1–6.

[159] Weickert, J. 1998. *Anisotropic Diffusion in Image Processing*. Teubner.

[160] Wenger, A., Gardner, A., Tchou, C., Unger, J., Hawkins, T., and Debevec, P. 2005. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Trans. on Graphics 24,* 3, 756–764.

[161] Werlberger, M., Pock, T., and Bischof, H. 2010. Motion Estimation with Non-Local Total Variation Regularization. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'10)*. IEEE Computer Society, 2464–2471.

[162] Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., and Bischof, H. 2009. Anisotropic Huber-L1 optical flow. In *Proc. of British Machine Vision Conference*. London, UK. to appear.

[163] Wilburn, B., Joshi, N., Vaish, V., Talvala, E.-V., Antunez, E., Barth, A., Adams, A., Horowitz, M., and Levoy, M. 2005. High Performance Imaging using large Camera Arrays. *ACM Trans. on Graphics 24,* 3, 765–776.

[164] Wilburn, B., Smulski, M., Lee, K., and Horowitz, M. A. 2002. The light field video camera. In *Proc. of SPIE Media Processors*. 29–36.

[165] Wilkes, L. 2009. The role of ocula in stereo post production. Tech. rep., The Foundry.

[166] Winnemöller, H., Olsen, S. C., and Gooch, B. 2006. Real-time video abstraction. *ACM Trans. on Graphics 25,* 3, 1221–1226.

[167] Wolberg, G. 1998. Image morphing: a survey. *The Visual Computer 14,* 8, 360–372.

[168] Wolf, M. 2006. Space, Time, Frame, Cinema: Exploring the Possibilities of Spatiotemporal Effects. *New Review of Film and Television Studies*, 369–374. www.digitalair.com/techniques/STFC.pdf.

[169] Würmlin, S., Lamboray, E., Waschbüsch, M., Kaufmann, P., Smolic, A., and Gross, M. 2005. Image-Space Free-Viewpoint Video. In *Proc. of Vision, Modeling and Visualization (VMV 2005)*. AKA, Heidelberg, 453–460.

[170] Xiao, J., Cheng, H., Sawhney, H., Rao, C., Isnardi, M., and Corporation, S. 2006. Bilateral filtering-based optical flow estimation with occlusion detection. In *Proc. of European Conference on Computer Vision (ECCV'06)*. Vol. 1. Springer, 211–224.

[171] Xiao, J. and Shah, M. 2004. Tri-View Morphing. *Computer Vision and Image Understanding 96*, 345–366.

[172] Xu, L., Chen, J., and Jia, J. 2008. A Segmentation Based Variational Model for Accurate Optical Flow Estimation. In *Proc. of European Conference on Computer Vision (ECCV'08)*. Springer, Berlin, 671–684.

[173] Yang, C., Duraiswami, R., Gumerov, N. A., and Davis, L. 2003. Improved fast gauss transform and efficient kernel density estimation. In *IEEE International Conference on Computer Vision (ICCV'03)*. IEEE Computer Society, Washington, DC, USA, 664–671.

[174] Yang, J. C., Everett, M., Buehler, C., and Mcmillan, L. 2002. A Real-Time distributed Light Field Camera. In *Proc. of Eurographics Rendering Workshop*. Eurographics Association, 77–86.

[175] Zach, C., Pock, T., and Bischof, H. 2007. A duality based approach for realtime TV-L^1 optical flow. In *Proc. of DAGM Symposium on Pattern Recognition*. Vol. 4713. 214–223.

[176] Zhang, Z., Wang, L., Guo, B., and Shum, H.-Y. 2002. Feature-based light field morphing. *ACM Trans. on Graphics 21,* 3, 457–464.

[177] Zheng, K. C., Colburn, A., Agarwala, A., Agrawala, M., Salesin, D., Curless, B., and Cohen, M. F. 2009. Parallax photography: creating 3d cinematic effects from stills. In *Proc. of Graphics Interface (GI'09)*. 111–118.

[178] Zilly, F., Müller, M., Eisert, P., and Kauff, P. 2010. The stereoscopic analyzer – an image-based assistance tool for stereo shooting and 3d production. In *Proc. of IEEE International Conference on Image Processing (ICIP'10)*. IEEE Computer Society, 4029–4032.

[179] Zimmer, H., Bruhn, A., Weickert, J., Valgaerts, L., Salgada, A., Rosenhahn, B., and Seidel, H.-P. 2009. Complementary optical flow. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, D.Cremers, Y.Boykov, A.Blake, and F.R.Schmidt, Eds. Lecture Notes in Computer Science, vol. 5681. 207–220.

[180] Zitnick, C., Jojic, N., and Kang, S. B. 2005. Consistent Segmentation for Optical Flow Estimation. In *IEEE International Conference on Computer Vision (ICCV'05)*. Vol. 2. 1308–1315.

[181] Zitnick, C., Kang, S., Uyttendaele, M., Winder, S., and Szeliski, R. 2004. High-Quality Video View Interpolation Using a Layered Representation. *ACM Trans. on Graphics 23,* 3, 600–608.

Curriculum Vitæ - Lebenslauf

Curriculum Vitæ

1979	born in Zweibrücken, Germany
1999	Highschool degree, main subjects mathematics, geography and chemistry
	Von der Leyen-Gymnasium, Blieskastel, Germany
1999 - 2005	Diploma in Computer Science
	Universität des Saarlandes, Saarbrücken, Germany
2005 - 2006	Ph.D. Student in Computer Science, Prof. M. Magnor
	Max-Planck-Institut für Informatik, Saarbrücken, Germany
2006 - 2010	Ph.D. Student Computer Science, Prof. M. Magnor
	TU Braunschweig, Germany

Lebenslauf

1979	geboren in Zweibrücken
1999	Allgemeine Hochschulreife
	Von der Leyen-Gymnasium Blieskastel
1999 - 2005	Diplom Informatik
	Universität des Saarlandes, Saarbrücken
2005 - 2006	Wissenschaftlicher Mitarbeiter, Prof. M. Magnor
	Max-Planck-Institut für Informatik, Saarbrücken
2006 - 2010	Wissenschaftlicher Mitarbeiter, Prof. M. Magnor
	TU Braunschweig

Abstract

Free-viewpoint video is a new form of visual medium that has received considerable attention in the last 10 years. Most systems reconstruct the geometry of the scene, thus restricting themselves to synchronized multi-view footage and Lambertian scenes.
This dissertation discusses a different approach and describes contributions to a purely image-based end-to-end system operating on sparse, unsynchronized multi-view footage. In particular, the discussion focuses on dense correspondence estimation and synthesis of in-between views. In contrast to previous approaches, the presented correspondence estimation technique is specifically tailored to the needs of image interpolation; the proposed multi-image interpolation technique advances the state-of-the-art by disposing the conventional blending step. Both algorithms are put to work in an image-based free-viewpoint video system and the author demonstrates their applicability to space-time visual effects production as well as to stereoscopic content creation.